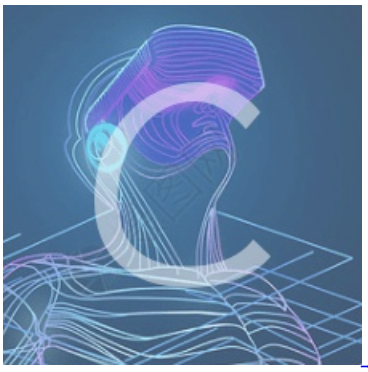
XCTF web unserialize3(反序列化漏洞)

原创

H4ppyD0g 于 2019-09-06 19:13:39 发布 293 收藏

分类专栏： CTF

本文链接：https://blog.csdn.net/weixin_42172261/article/details/100585079



php序列化就是将复杂的数据类型(比如说数组，字典等)转换为字符串，方便传输或者存库等操作,并且之后还能后恢复成原来的数据类型的过程。

serialize()	用于序列化对象或数组，并返回一个字符串。序列化对象后，可以很方便的将它传递给其他需要它的地方，且其类型和结构不会改变。
unserialize()	可以将serialize序列化复杂数据类型后得到的字符串类型的数据重新转换成原来复杂的数据类型。

魔术方法：php将所有以 __（两个下划线）开头的类方法保留为魔术方法。

serialize()和unserialize()函数对魔术方法的处理：

serialize()函数会检查类中是否存在魔术方法 **__sleep()** 。如果存在，该方法会先被调用，然后才执行序列化操作。

unserialize()函数会检查类中是否存在魔术方法 **__wakeup()**，如果存在，则会先调用 __wakeup 方法。

还有一些其它的魔术方法

__construc()	具有构造函数的类会在每次创建新对象时先调用此方法。
__destruct ()	析构函数会在到某个对象的所有引用都被删除或者当对象被显式销毁时执行。
__toString()	用于一个类被当成字符串时应怎样回应。

__wakeup()执行漏洞：一个字符串或对象被序列化后，如果其属性被修改，则不会执行__wakeup()函数，这也是一个绕过点。

本题代码为

```
class xctf{
public $flag = '111';
public function __wakeup(){
exit('bad requests');
}
?code=
```

判断应该是url传参code来绕过__wakeup函数，所以可以先获取序列化后的字符串，然后随便修改一个地方，再传参code，从而实现绕过。

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