PHP反序列化漏洞与CVE-2016-7124

30909889 / 2016-11-13 02:36:00 / 浏览数 5351 安全技术 漏洞分析 顶(0) 踩(0)

■■■http://www.cnbraid.com/2016/unserialize.html

0x01 背景

PHP存在着一个很有意思的漏洞，PHP对象注入，专业术语为反序列化漏洞。本篇文章将对php反序列化漏洞和CVE-2016-7124进行详细讲解。

0x02 序列化与反序列化概念

我们先要来了解一下什么是序列化和反序列化，所有php里面的值，我们都可以使用函数serialize()来返回一个包含字节流的字符串来表示。见下图：

序列化一个对象将会保存对象的所有变量，但是不会保存对象的方法，只会保存类的名字。所以对象A和对象B序列化后并没有什么区别。
unserialize()函数能够重新把字符串变回php原来的值。

0x03 反序列化漏洞

漏洞分析

漏洞的根源在于unserialize()函数的参数可控。如果反序列化对象中存在魔术方法，而且魔术方法中的代码有能够被我们控制，漏洞就这样产生了，根据不同的代码可以导致
漏洞代码如下：

```
<html>
<head>
<title>PHP■■■■</title>
</head>
<body>
<?php
class A{
   var $a = "test";
   function __destruct(){
       $fp = fopen("D:\\phpStudy\\WWW\\hello.php","w");
       fputs($fp,$this->a);
       fclose($fp);
   }
}
$test = $_POST['test'];
$test_unser = unserialize($test);
?>
</body>
</html>
```

这里使用了php的magic方法destruct（详情参考PHP: Magic Methods），而destruct是当一个对象被销毁时被自动调用的析构方法。
然后unserialize中参数可控，这样我们就可以构造一个序列化的对象A来控制其中的变量a的值，最终会产生漏洞。

漏洞证明

我们构造一个序列化的对象A并给其中变量a赋值为<?php phpinfo();?>如下：

可以看到成功将<?php phpinfo();?>写入hello.php。

实例分析

这里引用phithon师傅的文章：wecenter反序列化造成任意SQL语句执行
详情p牛已经分析过了，这里再次提下触发的几个点
1.首先是/system/aws_model.inc.php文件下

```
class AWS_MODEL
{
...
   /**
    * Model ■■■, ■■■■■■
       {
    */
   public function __destruct()
```

```
    {
        $this->master();
        foreach ($this->_shutdown_query AS $key => $query)
        {
            $this->query($query);
        }
    }
}
```

可以看到这里直接遍历了_shutdown_query对象，将其值传入query直接执行。
明显存在一个任意SQL语句执行漏洞，只要我生成一个AWS_MODEL类对象，再其销毁的时候就能执行任意SQL语句。
2.然后我们找到/app/m/weixin.php中使用了unserialize函数如下：

```
public function authorization_action()
{
    $this->model('account')->logout();
    unset(AWS_APP::session()->WXConnect);
    if (get_setting('weixin_account_role') != 'service')
    {
        H::redirect_msg(AWS_APP::lang()->_t('■■■■■■■■■■■■■■■■■■'));
    }
    else if ($_GET['code'] OR $_GET['state'] == 'OAUTH')
    {
        if ($_GET['state'] == 'OAUTH')
        {
            $access_token = unserialize(base64_decode($_GET['access_token']));
        }
```

上面这句话将$_GET['access_token']解码以后进行反序列化，所以可以构造特定的序列化对象AWS_MODEL来执行任意SQL语句了。

0x04 CVE-2016-7124

漏洞分析

CVE-2016-7124，简单来说就是当序列化字符串中表示对象属性个数的值大于真实的属性个数时会跳过__wakeup的执行，Demo如下：

```
<html>
<head>
<title>PHP■■■■</title>
</head>
<body>
<?php
class A{
    var $a = "test";
    function __destruct(){
        $fp = fopen("D:\\phpStudy\\WWW\\hello.php","w");
        fputs($fp,$this->a);
        fclose($fp);
    }
    function __wakeup()
        {
            foreach(get_object_vars($this) as $k => $v) {
                    $this->$k = null;
            }
            echo "Waking up...\n";
        }
}
$test = $_POST['test'];
$test_unser = unserialize($test);
?>
</body>
</html>
```

我们再次使用之前的payload测试，结果如下：

发现__wakeup函数成功执行，清除了对象属性，从而hello.php内容也为空了。

漏洞证明

我们将上面test=O:1:"A":1:{s:1:"a";s:18:"<?php phpinfo();?>";}中A的个数变成2或者大于2的数字如下：

test=O:1:"A":2:{s:1:"a";s:18:"<?php phpinfo();?>";}

然后再次执行发现绕过了__wakeup函数，成功将phpinfo()写入hello.php

参考来源：
http://bobao.360.cn/learning/detail/3091.html
https://bugs.php.net/bug.php?id=72663
wecenter反序列化造成任意SQL语句执行
延伸阅读：
SugarCRM v6.5.23 PHP反序列化 对象注入漏洞

点击收藏 | 0 关注 | 1
上一篇：- 下一篇：【独家连载】mysql注入天书（二...
1. 0 条回复
- 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板