New issue

Jump to bottom

There is a File upload vulnerability that can getshell #47

Open lavon321 opened this issue on Sep 21, 2020 · 0 comments

lavon321 commented on Sep 21, 2020 • edited

The file upload vulnerability here lies in the blacklist method used when verifying the suffix of the uploaded file. This verification method is not strict and is often bypassed by attackers in various ways

The PluginsUpload method in the application\service\PluginsAdminService.php file has a file creation operation, in which the input of the file_put_contents function is controllable

```
                    if($is_has_find == false)
                    {
                        continue;
                    }

                    // 截取文件路径
                    $file_path = substr($file, 0, strrpos($file, '/'));

                    // 路径不存在则创建
                    \base\FileUtil::CreateDir($file_path);

                    // 如果不是目录则写入文件
                    if(!is_dir($file))
                    {
                        // 读取这个文件
                        $file_size = zip_entry_filesize($temp_resource);
                        $file_content = zip_entry_read($temp_resource, $file_size
                        @file_put_contents($file, $file_content);
                    }

                    // 关闭目录项
                    zip_entry_close($temp_resource);
                }
```

Line 1072 checks the file suffix name, here is the blacklist check

```
        $is_has_find = false;
        foreach($dir_list as $dir_key=>$dir_value)
        {
            if(strpos($file, $dir_key) !== false)
            {
                // 仅插件目录模块支持php文件
                if($dir_key != '_controller_')
                {
                    // 排除后缀文件
                    $pos = strripos($file, '.');
                    if($pos !== false)
                    {
                        if(in_array(substr($file, $pos), self::$exclude_ext))
                        {
                            continue;
                        }
                    }
                }

                // 匹配成功文件路径处理、跳出循环
                $file = str_replace($plugins_name.'/'.$dir_key.'/', '', $dir_value.$file
                $is_has_find = true;
                break;
```

The value in the private static variable $exclude_ext is '.php', which can easily be bypassed

```
use app\service\SqlconsoleService;

/**
 * 应用管理服务层
 * @author   Devil
 * @blog     http://gong.gg/
 * @version  0.0.1
 * @datetime 2016-12-01T21:51:08+0800
 */
class PluginsAdminService
{
    // 排除不能使用的名称
    public static $plugins_exclude_verification = ['view', 'shopxo', 'www'];

    // 排除的文件后缀
    private static $exclude_ext = ['.php'];

    /**
     * 列表
     * @author   Devil
     * @blog     http://gong.gg/
     * @version  1.0.0
     * @date     2018-09-29
```

There are many ways to bypass the blacklist verification of suffix names. Taking my local Windows system environment as an example, you can upload file names that do not conform to the Windows file naming rules

> shell.php::$DATA
> shell.php::$DATA.....
> shell.php.
> shell.php(空格)
> shell.php:1.jpg

The windows system will automatically remove the content behind the symbols that do not conform to the rules. You can change the file suffix in the linux environment and upload it to the website
Through the audit of the PluginsUpload method in the PluginsAdminService.php file
When the zip archive does not match the resource directory, it will jump out of the loop of reading the archive file

```
                // 没有匹配到则直接跳过
                if($is_has_find == false)
                {
                    continue;
                }

            // 仅控制器模块支持php文件
            if($dir_key != '_controller_')
            {
                // 排除后缀文件
                $pos = strripos($file, '.');
                if($pos !== false)
                {
                    if(in_array(substr($file, $pos), self::$exclude_ext))
                    {
                        continue;
                    }
                }
            }

            // 匹配成功文件路径处理，跳出循环
            $file = str_replace($plugins_name.'/'.$dir_key.'/', '', $dir_value.$file);
            $is_has_find = true;
            break;
        }
    }

    // 模块目录
    $dir_list = [
        '_controller_'      => APP_PATH.'plugins'.DS,
        '_view_'            => APP_PATH.'plugins'.DS.'view'.DS,
        '_css_'             => ROOT.'public'.DS.'static'.DS.'plugins'.DS.'css'.DS,
        '_js_'              => ROOT.'public'.DS.'static'.DS.'plugins'.DS.'js'.DS,
        '_images_'          => ROOT.'public'.DS.'static'.DS.'plugins'.DS.'images'.DS,
        '_uploadfile_'      => ROOT.'public'.DS.'static'.DS.'upload'.DS.'file'.DS,
        '_uploadimages_'    => ROOT.'public'.DS.'static'.DS.'upload'.DS.'images'.DS,
        '_uploadvideo_'     => ROOT.'public'.DS.'static'.DS.'upload'.DS.'video'.DS,
    ];
```

And the resource directory cannot be *controller*, because the directory corresponding to *controller* exists, and the compressed package will be closed directly without entering the subsequent file writing operation

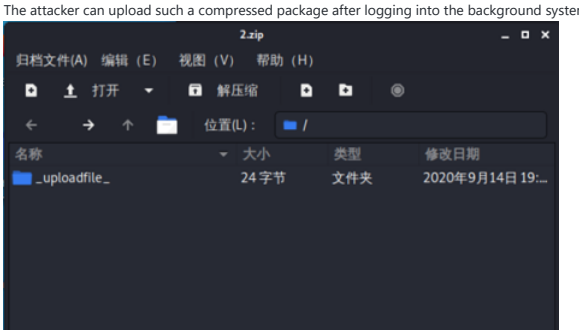
```
            // 应用是否存在
            $ret = self::PluginsExist($plugins_name);
            if($ret['code'] != 0)
            {
                zip_entry_close($temp_resource);
                return $ret;
            }
```

Finally, the method is called in the Upload method of the application\admin\controller\Pluginsadmin.php file

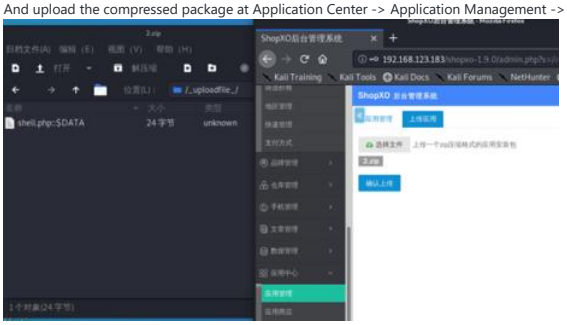
```
    public function Upload()
    {
        // 是否ajax
        if(!IS_AJAX)
        {
            return $this->error('非法访问');
        }

        // 开始处理
        return PluginsAdminService::PluginsUpload(input());
    }
```

The attacker can upload such a compressed package after logging into the background system



And upload the compressed package at Application Center -> Application Management -> Upload Application



Visit public\static\upload\file_uploadfile_\shell.php

In application\service\ThemeService.php there is also the same blacklist verification problem for uploaded files

```
class ThemeService
{
    // 静态目录和html目录
    private static $html_path = 'application'.DS.'index'.DS.'view'.DS;
    private static $static_path = 'public'.DS.'static'.DS.'index'.DS;

    // 排除的文件后缀
    private static $exclude_ext = ['.php'];

    /**
     * 获取模板列表
     * @author  Devil
     * @blog    http://gong.gg/
     * @version 0.0.1
```

The processing logic is very similar to the above file

After logging in to the system, upload the zip archive at the site management -> theme management -> theme installation

Visit public\static\index\test.php after uploading

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant