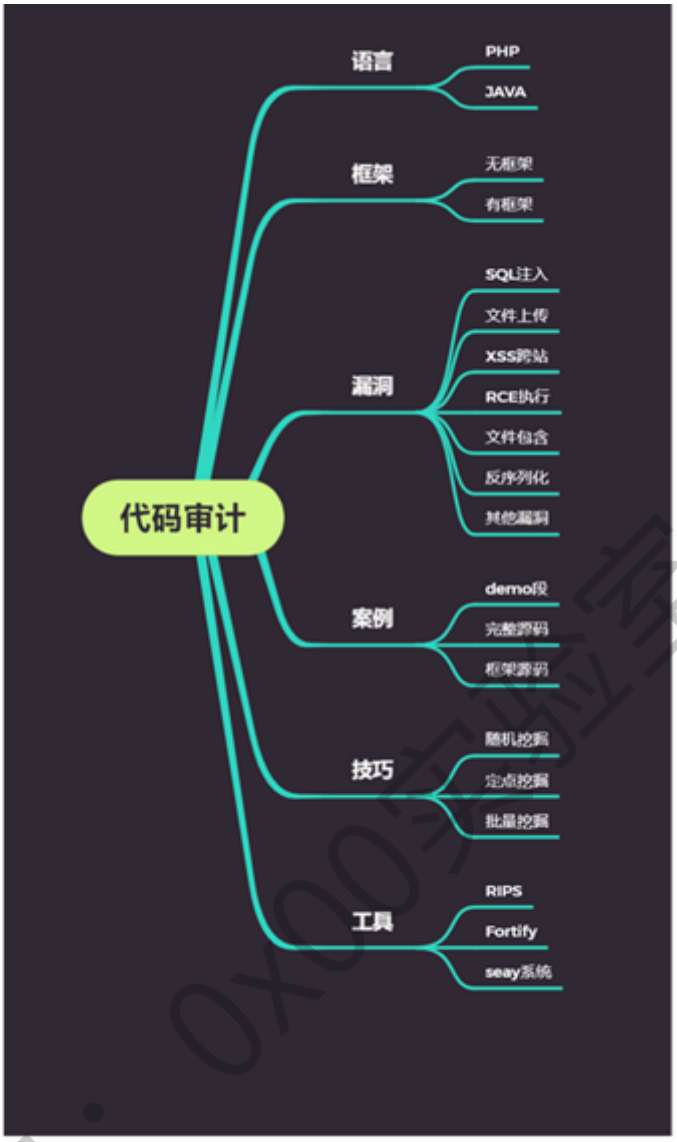


知识点1:

关键字搜索:(函数，键字，全局变量等)
文件上传，$_FILES,move_uploaded_file等

应用功能抓包:(任何可能存在上传的应用功能点)前台会员中心，后台新闻添加等可能存在上传的地方

没有文件后缀 判断MIME类型

知识点2 :

MVC开发框架类: https 😕 / www.cnblogs.com/ wsybky/p/8638876.html
断点调试，phpstorm不需要看懂代码，只需跟着跳点走。

知识点3 :

Thinkphp框架:https: / / sites.thinkphp.cn/1556331

演示案例:

PHP文件上传全局变量$_FILES
PHPStorm+xdebu断点调试演示
Beescms无框架后台任意文件上传
Finecms基于前台MVC任意文件上传

搜索$_FILES->后台中心->上传图像->跟踪代码->逻辑判断
业务功能分析-→>会员中心->上传图像->跟踪代码-→>逻辑判断
搜索文件上传->会员中心->上传图像->跟踪代码->逻辑判断

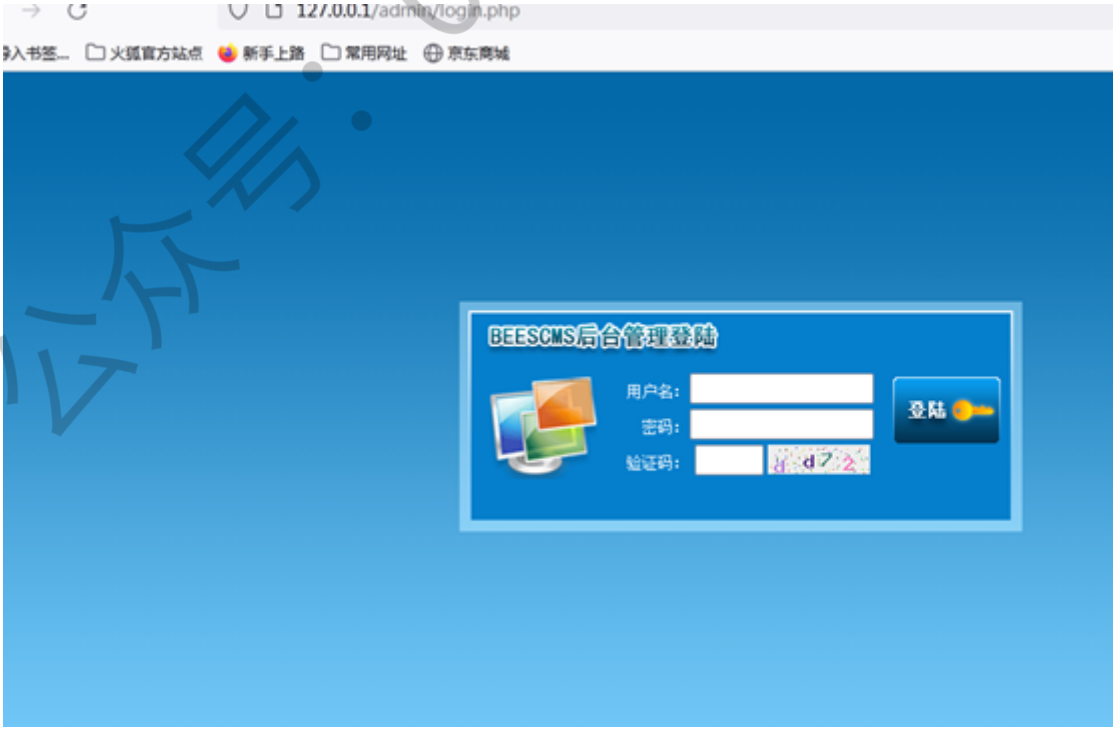
52

漏洞关键字

SQL注入：
select insert update mysql_query mysqli等
文件上传：
$*FILES,type="file"*，上传，*move_upload_file()*等
*XSS*跨站：
*print print_r echo sprintf die var_dump var_export*等
文件包含：
*Include include_once require require_once*等
代码执行：
*eval assert preg replace call user func call user func array*等
命令执行：
system exec shell_exec ``passthru pcntl_exec popen proc_open
变量覆盖：
*extract() parse_str() importrequestvariables() $$*等
反序列化：
*serialize() unserialize() _construct _destruct*等
通用关键字：
*$*GET $*POST $*REQUEST $*FILES $*SEVER

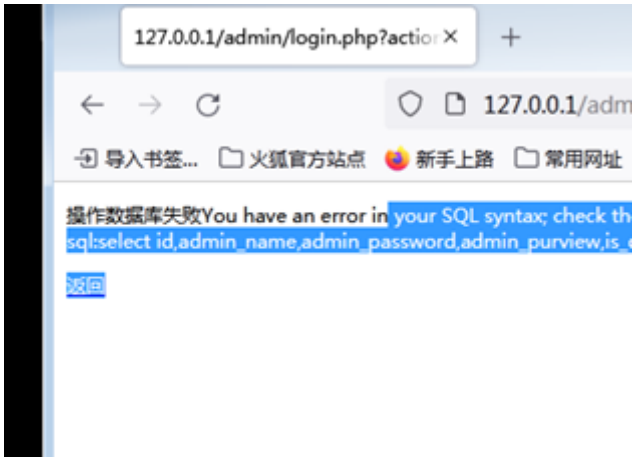
这里插入的是我的复现beescms过程

\1. 首先，访问本地搭建的靶场



登录后台，给账号加入特殊符号

Admin‘



这里报错了，显然这里有显错注入了

我们找到代码的判断登录出，也就是login.php文件

```
global $submit,$user,$password,$_sys,$code;
$submit=$_POST['submit'];
$user=fl_html(fl_value($_POST['user']));
$password=fl_html(fl_value($_POST['password']));
$code=$_POST['code'];
if(!isset($submit)){
```

也就是这一部分！

```
}
//判断登录
elseif($action=='ck_login'){
    global $submit,$user,$password,$_sys,$code;
    $submit=$_POST['submit'];
    $user=fl_html(fl_value($_POST['user']));
    $password=fl_html(fl_value($_POST['password']));
    $code=$_POST['code'];
    if(!isset($submit)){
```

标记的这里是声明了全局变量，其中包括用户名和密码

```
global $submit,$user,$password,$_sys,$code;
$submit=$_POST['submit'];
$user=fl_html(fl_value($_POST['user']));
$password=fl_html(fl_value($_POST['password']));
$code=$_POST['code'];
if(!isset($submit)){
```

这里能看到是用户和密码的部分，这里能够看得出定义了两个函数来对用户和密码判断

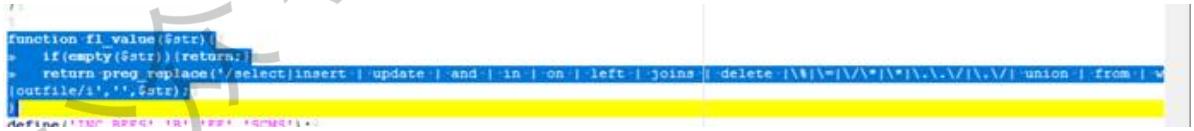
```
$submit=$_POST['submit'];
$user=fl_html(fl_value($_POST['user']));
$password=fl_html(fl_value($_POST['password']));
$code=$_POST['code'];
if(!isset($submit)){
    msg('请从登陆页面进入');
}
```

就是这俩

选中fl_html进行函数定位

这是结果，点进去看一下!

这俩函数都在fun.php这个文件中



fl_value

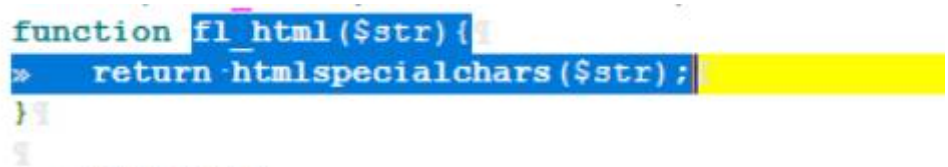
这个函数一看就知道他是过滤关键词的...还有符号!

就是这些，经查阅度娘，这个函数他就过滤一次

这意味着这里是可以用双写绕过

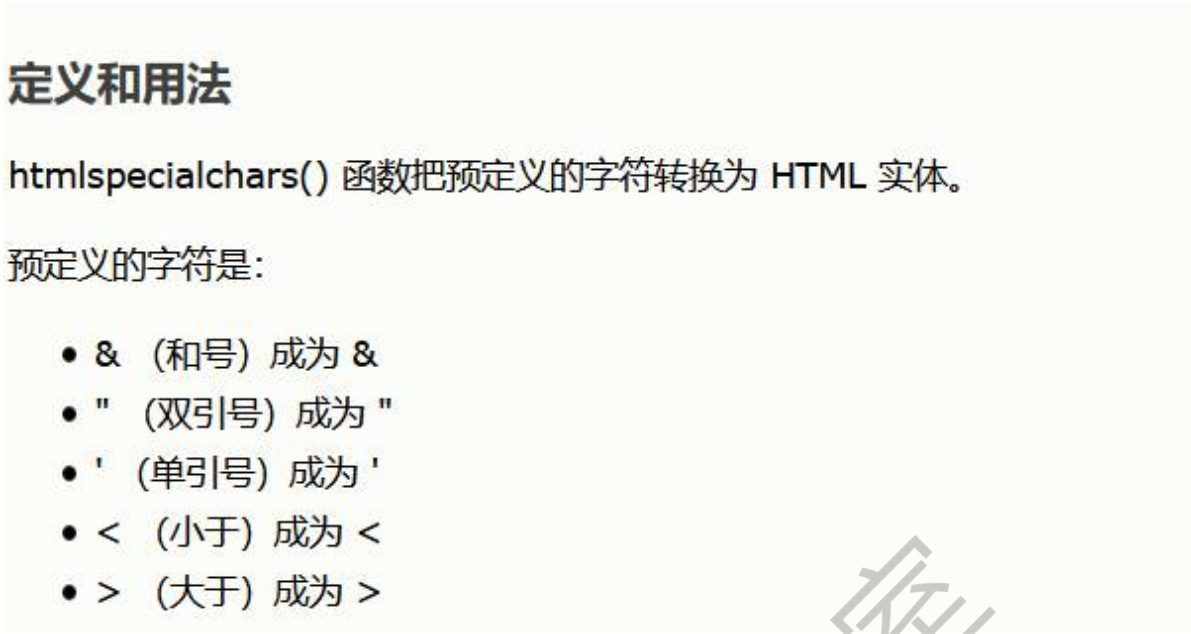
fl_html

这个函数看这里



他的意思是调用了**htmlspecialchars**($str)这个东西

这是个啥啊？问下度娘



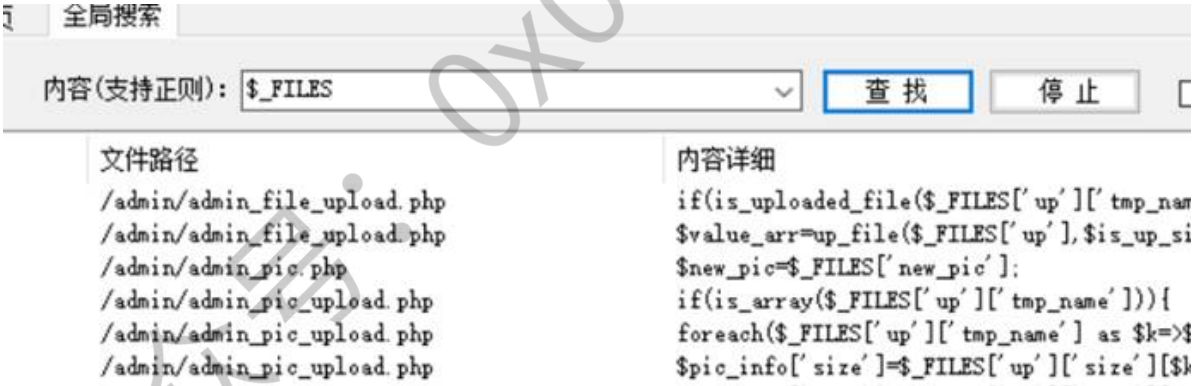
好吧，不太懂！这里继续问度娘，最后得出答案大概是这样的

就是说'转完还是'，然后就造成了一个报错注入，就是说

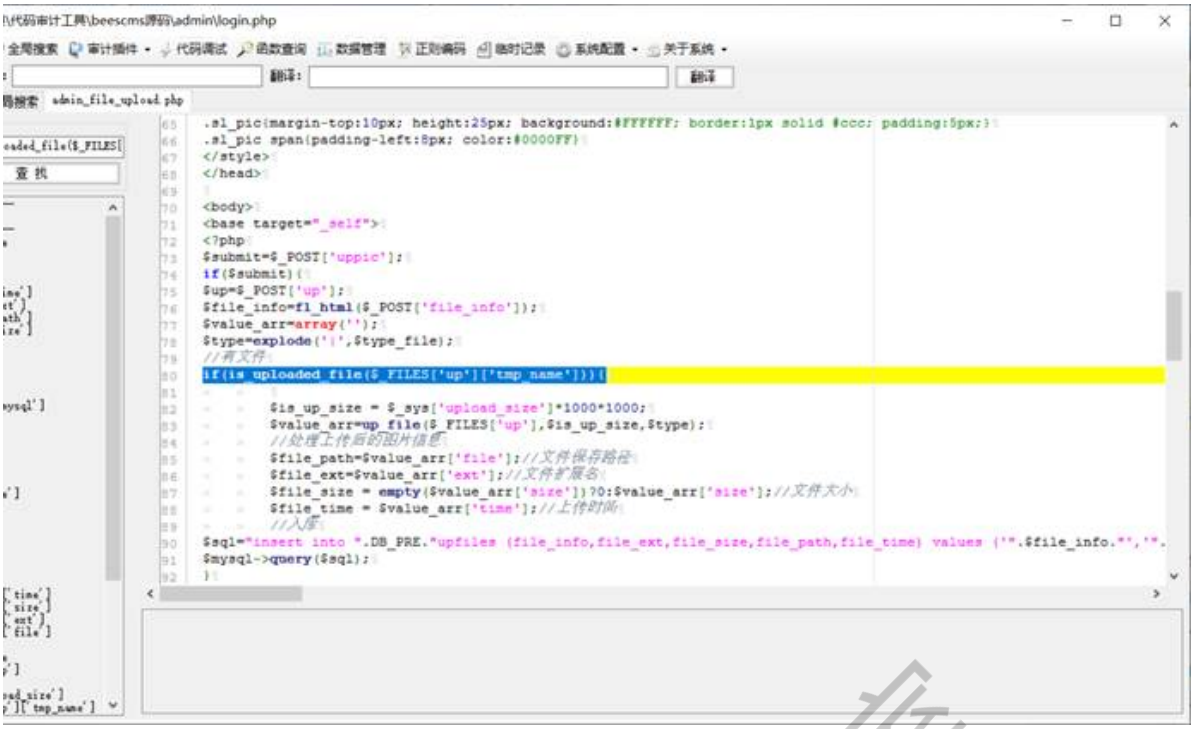
这个漏洞加上之前的双写就可以进行sql注入了

接下来接着审计

搜一下文件上传的函数$_FILES



我们选第一个，进去后就是这个样子



然后我们配合抓包来审计

首先，这个文件上传在 admin_file_upload.php 这个文件中，我们先访问一下

然后随便上传个文件，抓包看看

```
POST /beescms/admin/admin_file_upload.php HTTP/1.1
Host: 127.0.0.1:8082
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1

Content-Disposition: form-data; name="file_info"


-----------------------------1034979923142
Content-Disposition: form-data; name="uppic"
```

这里有两个参数指向admin_file_upload.php

跟进这两个参数，先是uppic

就是这样

```
$submit=$_POST['uppic'];
if($submit){
$up=$_POST['up'];
$file_info=fl_html($_POST['file_info']);
$value_arr=array('');
$type=explode('|',$type_file);
//有文件
if(is_uploaded_file($_FILES['up']['tmp_name'])){
    $is_up_size = $_sys['upload_size']*1000*1000;
```

这里大概的意思是将`$_POST['uppic'];`的值给`$submit`

然后判断`$submit)`的值

```
$up=$_POST['up'];
$file_info=fl_html($_POST['file_info']);
```

将后者的值赋给前者

```
$value_arr=array('');
$type=explode('|',$type_file);
//有文件
if(is_uploaded_file($_FILES['up']['tmp_name'])){
    $is_up_size = $_sys['upload_size']*1000*1000;
    $value_arr=up_file($_FILES['up'],$is_up_size,$type);
    //处理上传后的图片信息
    $file_path=$value_arr['file'];//文件保存路径
    $file_ext=$value_arr['ext'];//文件扩展名
    $file_size = empty($value_arr['size'])?0:$value_arr['size'];//文件大小
    $file_time = $value_arr['time'];//上传时间
    //入库
```

接收文件

```
if(is_uploaded_file($_FILES['up']['tmp_name'])){
```

判断文件是否被接受，用`($_FILES['u`接受

```
$is_up_size = $_sys['upload_size']*1000*1000;
```

这行是控制接收片的大小

```
$file_size = empty($value_arr['size'])?0:$value_arr['size'];//文件大小
$file_time = $value_arr['time'];//上传时间
```

这些事判断是否为空，获取文件大小，获取上传时间

```
if(is_uploaded_file($_FILES['up']['tmp_name'])){
        $is_up_size = $_sys['upload_size']*1000*1000;
        $value_arr=up_file($_FILES['up'],$is_up_size,$type);
        //处理上传后的图片信息
        $file_path=$value_arr['file'];//文件保存路径
        $file_ext=$value_arr['ext'];//文件扩展名
        $file_size = empty($value_arr['size'])?0:$value_arr['size'];//文件大小
        $file_time = $value_arr['time'];//上传时间
        //入库
$sql="insert into ".DB_PRE."upfiles (file_info,file_ext,file_size,file_path,file_time) values ('".$file_info."','".
$mysql->query($sql);
}
```

大体就是这个样子，后面进入了一个sql语句的代码

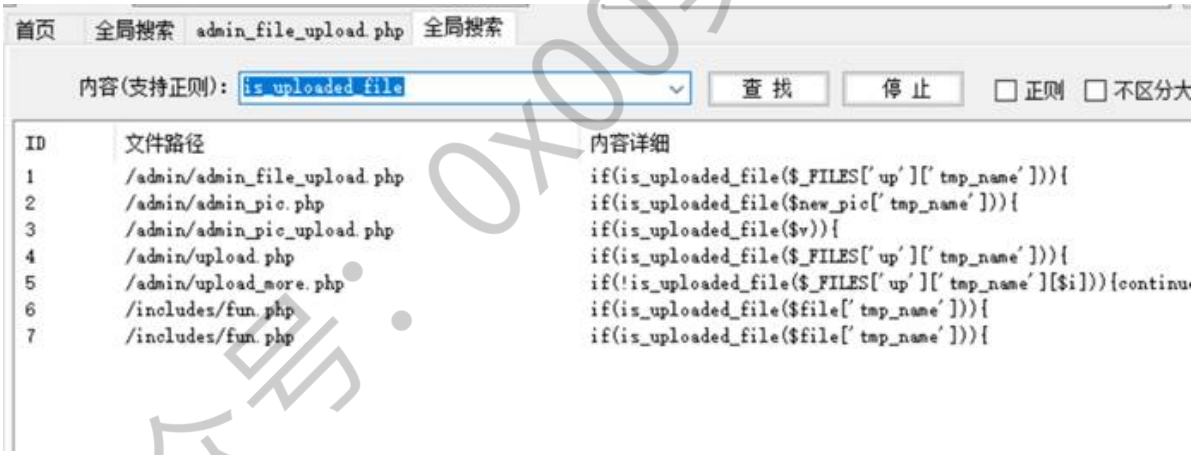
然后我们跟进我们不太清楚的函数，就是上图标记的那个

确定不是定位函数后问了度娘

> php中的is_uploaded_file()函数是函数判断指定的文件是否是通过 HTTP POST 上传的。
>
> 判断文件是否是通过HTTPPOST上传的。如果 filename 所给出的文件是通过 HTTP POST 上传的则返回 TRUE。这可以用来确保恶意的用户无法欺骗脚本去访问本不能访问的文件，例如 /etc/passwd。

也就是说这个函数就是检测图片能否上传的

接着搜索



这里我们挨个寻找，一直找到这里

```php
.oad.php 全局搜索 fun.php
567  *上传图片
568  *$url=远程图片,$file-上传文件,$size-允许大小,$type-上传文件类型,$thumb-缩略图,$thumb_width-缩略图宽度,
569  *return $arr-原始图和缩略图
570  */
571  function up_img($file,$size,$type,$thumb=0,$thumb_width='',$thumb_height='',$logo=1,$pic_alt=''){
572      if(file_exists(DATA_PATH.'sys_info.php')){include(DATA_PATH.'sys_info.php');}
573      if(is_uploaded_file($file['tmp_name'])){
574      if($file['size']>$size){
575          msg('图片超过'.$size.'大小');
576      }
577      $pic_name=pathinfo($file['name']);//图片信息
578
579      $file_type=$file['type'];
580      if(!in_array(strtolower($file_type),$type)){
581          msg('上传图片格式不正确');
582      }
583      $path_name="upload/img/";
584      $path=CMS_PATH.$path_name;
585      if(!file_exists($path)){
586          @mkdir($path);
587      }
588      $up_file_name=empty($pic_alt)?date('YmdHis').rand(1,10000):$pic_alt;
589      $up_file_name2=iconv('UTF-8','GBK',$up_file_name);
590      $file_name=$path.$up_file_name2.'.'.$pic_name['extension'];
591
592      if(file_exists($file_name)){
593          msg('已经存在该图片，请更改图片名称！');//判断是否重名
594      }
```

这里我们已经能看出这是检测我们上传的图片的代码了

```php
    $pic_name=pathinfo($file['name']);//图片信息

    $file_type=$file['type'];
    if(!in_array(strtolower($file_type),$type)){
        msg('上传图片格式不正确');
    }
    $path_name="upload/img/";
```

、

这里有判断，猜测是白名单，查询下这个函数

差不多，应该就是了，她后面那个函数也查询一下

是这样的，那这句话的意思就明白了，

对文件`$type)){`进行白名单检测，那这个文件里的内容就是

```
允许上传的文件类型:zip|gz|rar|iso|doc|xsl|ppt|wps|swf|mpg|mp3|rm|rmvb|wmv|wma|wav|mid|mov)
```

这些

`$file_type`对这个变量跟进

```
$pic_name=pathinfo($file['name']);
$file_type=$pic_name['extension'];
```

这里就需要对函数`pathinfo(`查询

定义和用法

pathinfo() 函数以数组的形式返回关于文件路径的信息。

返回的数组元素如下：

- [dirname]: 目录路径
- [basename]: 文件名
- [extension]: 文件后缀名
- [filename]: 不包含后缀的文件名

```php
<?php
print_r(pathinfo("/testweb/test.txt"));
?>
```

上面的代码将输出：

```
Array
(
    [dirname] => /testweb
    [basename] => test.txt
    [extension] => txt
    [filename] => test
)
```

这里就得到了一个比较关键的 `[extension] => txt` 后缀名，

那么就可以知道，用 `($file_type` 与 `$type)` 比对后缀名

这边上传的代码就差不多审核完了，经过抓包拦截，改content-type一系列操作，然后，没有漏洞
emmmmmmmmm

继续寻找后发现这个地方还有一个上传点，我们找到他的代码部分

admin_pic_upload.php

```
$thumb_width=$_POST['thumb_width'];
$thumb_height=$_POST['thumb_height'];
$up_is_thumb=intval($is_thumb);
$up_thumb_width=empty($thumb_width)?$_sys['thump_width']:intval($thumb_width);
$up_thumb_height=empty($thumb_height)?$_sys['thump_height']:intval($thumb_height);
$pic_cate=$_POST['pic_cate'];
if(is_array($_FILES['up']['tmp_name'])){
foreach($_FILES['up']['tmp_name'] as $k=>$v){
if(empty($v)){continue;}
$value_arr=array('');
$pic_info=array('');
//有图上传图片
if(is_uploaded_file($v)){
        $pic_info['tmp_name']=$v;
        $pic_info['size']=$_FILES['up']['size'][$k];
        $pic_info['type']=$_FILES['up']['type'][$k];
        $pic_info['name']=$_FILES['up']['name'][$k];
        $pic_name_alt=empty($is_alt)?'':$pic_alt[$k];
        $is_up_size = $_sys['upload_size']*1000*1000;
        $value_arr=up_img($pic_info,$is_up_size,array('image/gif','image/jpeg','image/png','image/j
        //处理上传后的图片信息
        $pic_name=$value_arr['up_pic_name'];//图片名称空
        $pic_ext=$value_arr['up_pic_ext'];//图片扩展名
        $pic_title = $pic_alt[$k];//图片描述
        $pic_size = $value_arr['up_pic_size'];//图片大小
        $pic_path = $value_arr['up_pic_path'];//上传路径
        $pic_time = $value_arr['up_pic_time'];//上传时间
        $pic_thumb = iconv('GBK','UTF-8',$value_arr['thumb']);//缩略图
```

在这个地方，审核方法和之前一样，然而这个地方

```
$pic_info['name']=$_FILES['up']['name'][$k];
$pic_name_alt=empty($is_alt)?'':$pic_alt[$k];
$is_up_size = $_sys['upload_size']*1000*1000;
$value_arr=up_img($pic_info,$is_up_size,array('i
//处理上传后的图片信息
$pic_name=$value_arr['up_pic_name'];//图片名称空
$pic_ext=$value_arr['up_pic_ext'];//图片扩展名
$pic_title = $pic_alt[$k];//图片描述
$pic_size = $value_arr['up_pic_size'];//图片大小
$pic_path = $value_arr['up_pic_path'];//上传路径
$pic_time = $value_arr['up_pic_time'];//上传时间
```

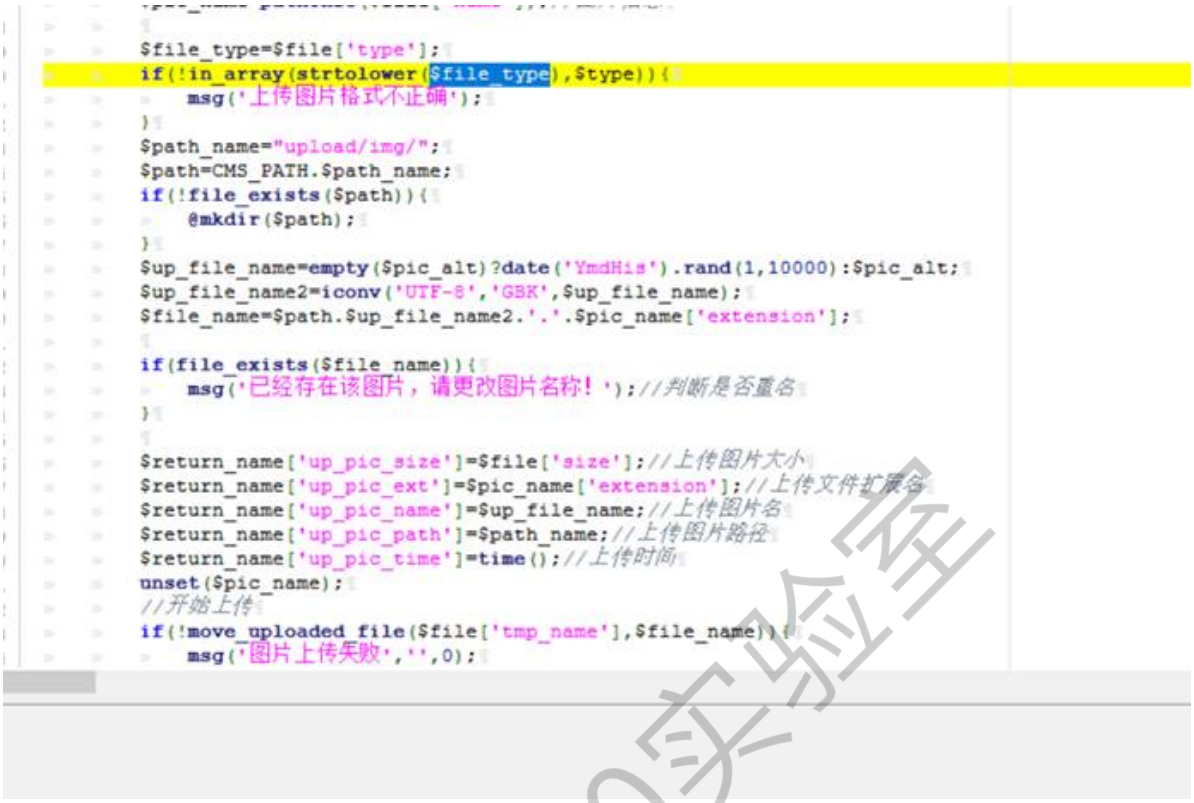
标记的函数是判断文件上传的函数，定位这个函数

```
function up_img($file,$size,$type,$thumb=0,$thumb_width='',$thumb_height='',$logo=1,$pic_alt='')
    if(file_exists(DATA_PATH.'sys_info.php')){include(DATA_PATH.'sys_info.php');}
    if(is_uploaded_file($file['tmp_name'])){
    if($file['size']>$size){
        msg('图片超过'.$size.'大小');
    }
    $pic_name=pathinfo($file['name']);//图片信息

    $file_type=$file['type'];
    if(!in_array(strtolower($file_type),$type)){
        msg('上传图片格式不正确');
    }
    $path_name="upload/img/";
    $path=CMS_PATH.$path_name;
    if(!file_exists($path)){
        @mkdir($path);
    }
    $up_file_name=empty($pic_alt)?date('YmdHis').rand(1,10000):$pic_alt;
    $up_file_name2=iconv('UTF-8','GBK',$up_file_name);
    $file_name=$path.$up_file_name2.'.'.$pic_name['extension'];

    if(file_exists($file_name)){
        msg('已经存在该图片，请更改图片名称！');//判断是否重名
    }
```

```
$file_type=$file['type'];
if(!in_array(strtolower($file_type),$type)){
    msg('上传图片格式不正确');
```

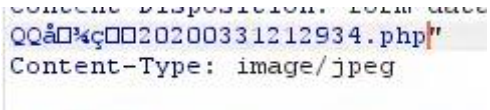
这里还是跟之前一样是白名单检测，但是

他只是检测了mime类型，并没有对文件后缀名判断



后面有上传操作，没有对后缀判断

也就是说这个地方验证通过就可以了



直接改成php就可以上传了！

功能点或者关键字分析可能存在漏洞
抓包或者搜索关键字找到代码出处对应文件
追踪过滤或者接受的数据函数，寻找出发此函数或者代码的地方进行触发测试

xhcms-无框架-文件包含跨站-搜索或应用-include

通过应用及URL地址等分析可能存在xss及包含安全
抓包找到xSS无过滤代码块及文件包含有后缀需绕过代码块
unlink，delfile是php中对应删除的函数
删除数据库安装文件，可以重装数据库。

53

漏洞关键字

SQL注入：
select insert update mysql_query mysqli等
文件上传：
$FILES,type="file"，上传，*move_upload_file()*等
*XSS*跨站：
*print print_r echo sprintf die var_dump var_export*等
文件包含：
*Include include_once require require_once*等
代码执行：
*eval assert preg replace call user func call user func array*等
命令执行：
system exec shell_exec ``passthru pcntl_exec popen proc_open
变量覆盖：
*extract() parse_str() importrequestvariables() $$*等
反序列化：
*serialize() unserialize() _construct _destruct*等
通用关键字：
*$*GET $*POST $*REQUEST $*FILES $*SEVER

双$$可能存在变量覆盖漏洞

foreach遍历PHP中array（数组）中的值 var_dump()输出数组的变量

```
foreach(array('_COOKIE', '_POST', '_GET') as $_request) {
    foreach($$_request as $_key => $_value) {
        $_key{0} != '_' && $$_key = daddslashes($_value);
    }
}
```

变量名等于参数名，实现覆盖，接受*$GET,$*POST,$_COOKIE的值
配合文件包含->满足包含变量覆盖的配置文件->找到可控变量包含的代码段->将可控变量进行覆盖->造成包含一句话木马

变量覆盖配合文件包含实现任意文件包含
自动审计或搜索关键字找到文件及代码段，全局配置文件搜索或访问触发全局配置文件配合手写代码测试变量覆盖
配合文件包含漏洞覆盖指定文件实现文件包含攻击获取权限

追踪$module变量出处文件，然后分析如何覆盖它达到目的
payload:/about/index.php?fmodele=7&module=1.txt

反序列化

自动审计或搜索关键字找到文件及代码段
wakeup（） //使用unserialize时触发
sleep（） //使用serizalize时触发
destruct() //对象被销毁时触发
call() //在对象上下文中调用不可访问的方法时触发
callstatic() //在静态上下文中调用不可访问的方法时触发
get() //用于从不可访问的属性读取数据
__isset() //在不可访问的属性上调用iisset（）或者empyt() 触发

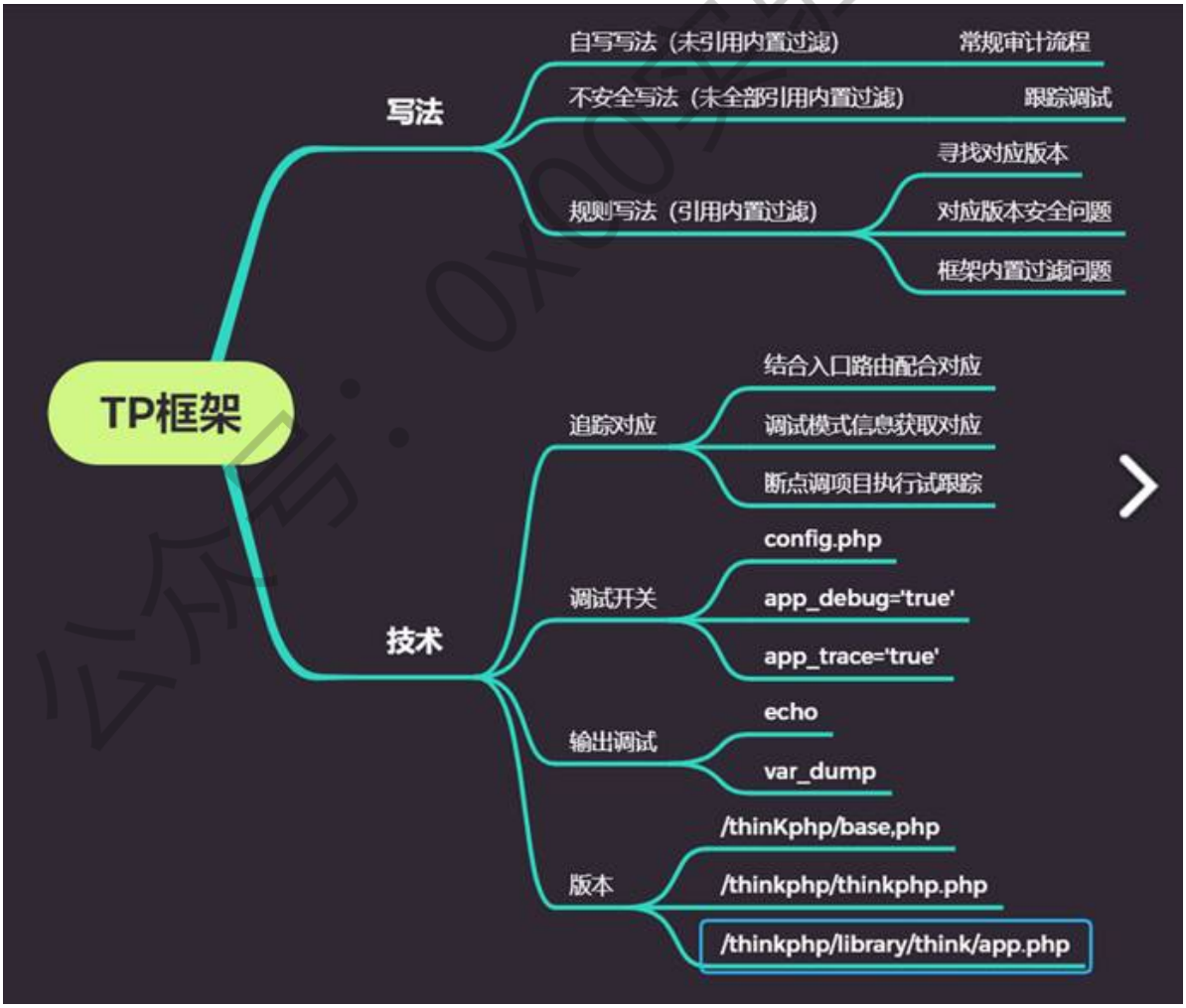
找到__wakeup代码段，代码段调用函数中存在eval（）等函数操作，可调试payload
构造getsource（）利用payload，实现file_get_contents函数配合eval执行
payload：action=test&configuration=0:10:"PMA_config"：1：{s:6："source",11:"d:/test.txt"；}

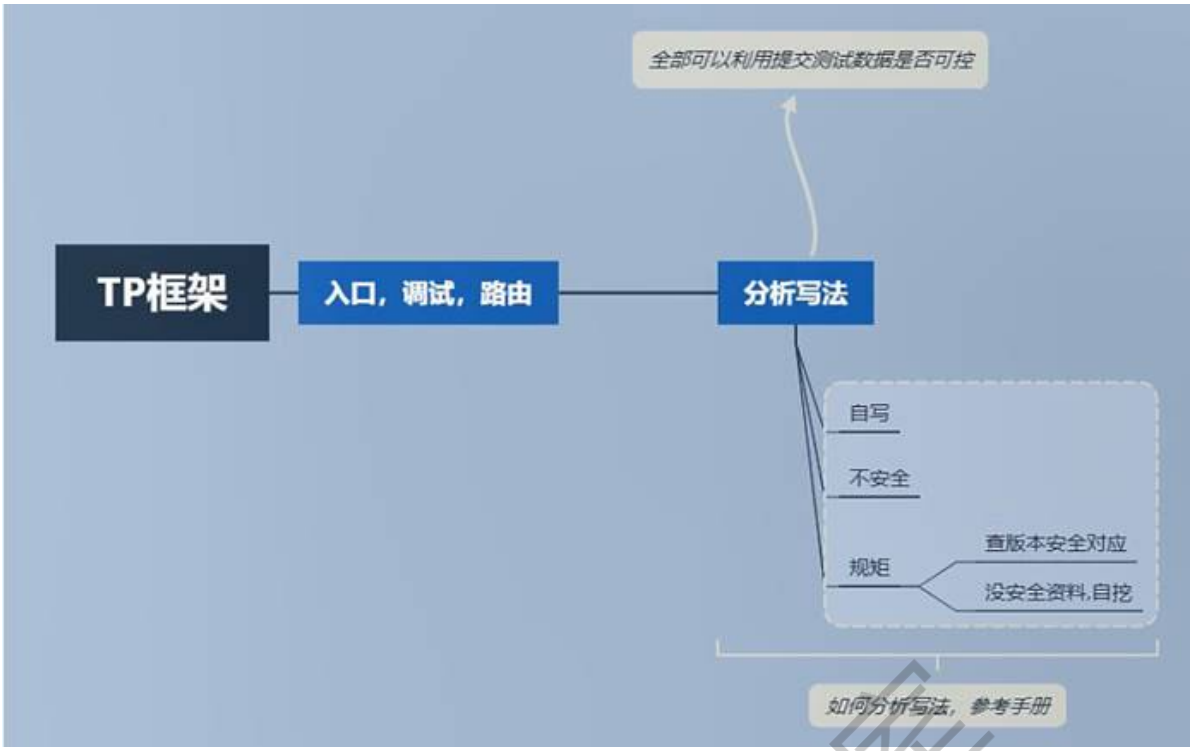
Thinkphp5 简要知识点

模块/目录/文件/方法

入口文件，调试模式，访问路由，访问对应
内置安全等测试访问不同方法名，不同文件不同方法名
测试常规写法SQL注入，TP5规定写法SQL注入

thinkphp框架





知识点：

调试，访问，路由，配置，版本等

自写写法，不安全写法，规则写法

调试模式信息获取对应结合入口路由配合对应项目断点调试执行跟着