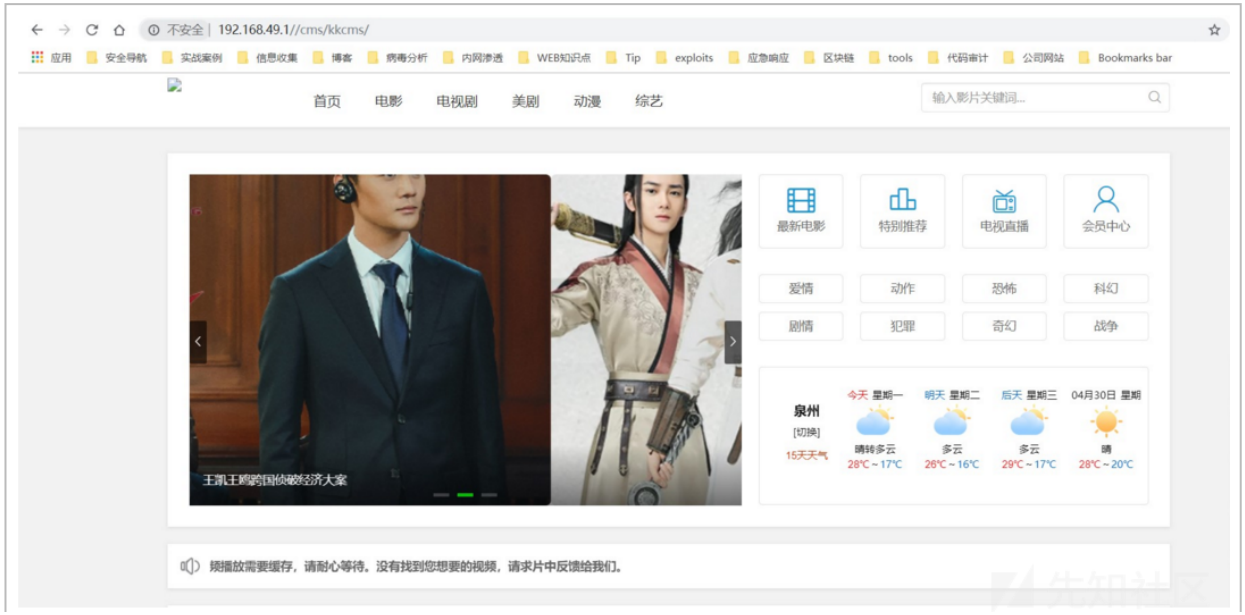
入门 KKCMS 代码审计 - 先知社区

> 先知社区，先知安全技术社区

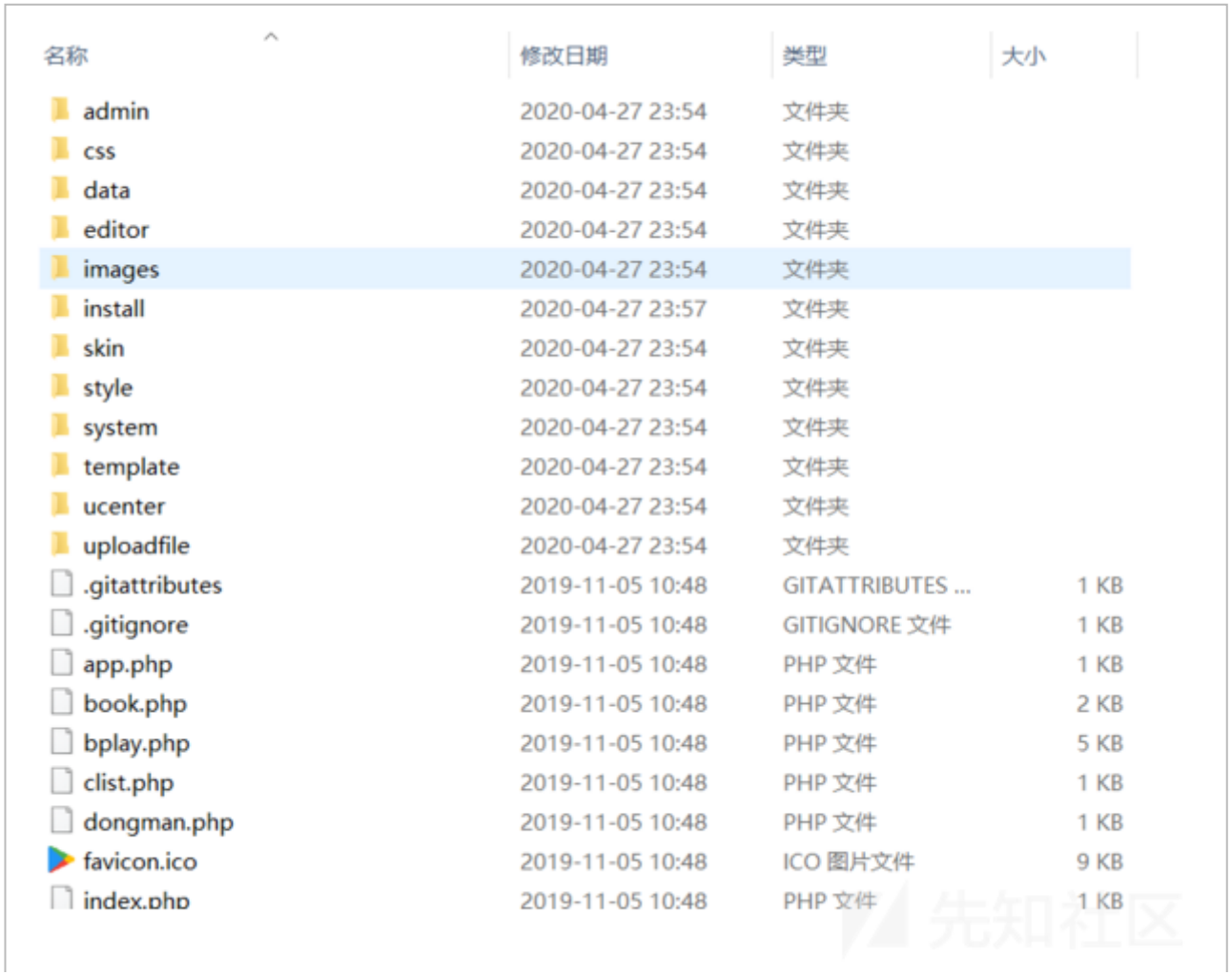
一、安装

安装过程不再赘述，安装后打开站点如下图所示
按 CMS 的介绍可以用来发卡或者来采集影视进行播放
由于是新手导向，所以可能会比较详细（入门）



(https://xzfile.aliyuncs.com/media/upload/picture/20200429150315-7f147de6-89e7-1.png)

其中部分文件



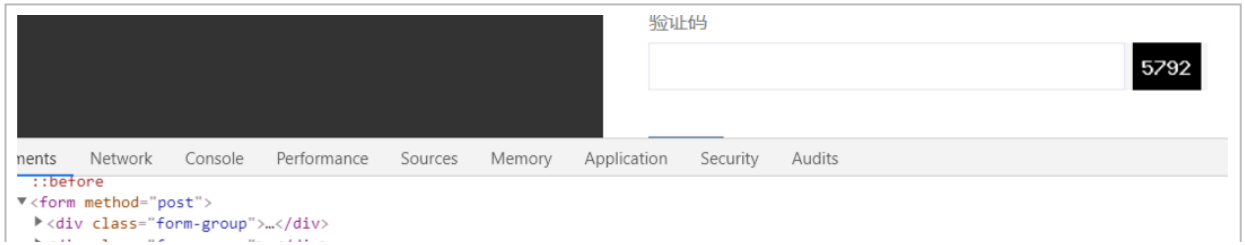
(https://xzfile.aliyuncs.com/media/upload/picture/20200429150325-8501afda-89e7-1.png)

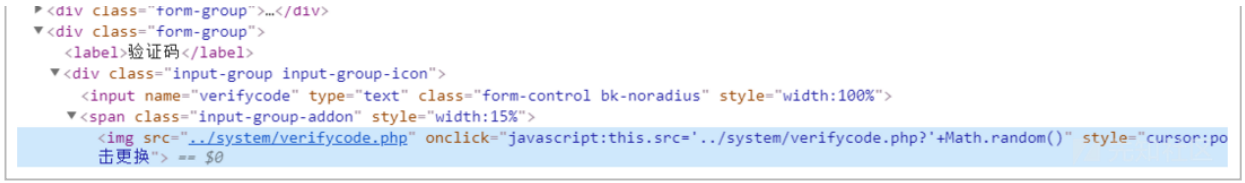
二、验证码重用

对比 session 和传入的 verifycode 是否相等

(https://xzfile.aliyuncs.com/media/upload/picture/20200429150910-5332a0da-89e8-1.png)

每次失败访问后，都会进行刷新跳转，然后重新执行一次 JS 代码，但是由于 Burp 默认不解析 js，所以这里存在验证码复用的漏洞





(https://xzfile.aliyuncs.com/media/upload/picture/20200429150922-59ee82cc-89e8-1.png)

搜索此参数，发现只在 verifycode.php 中才存在 $_SESSION['verifycode']
生成 0——9 的随机 4 位验证码。

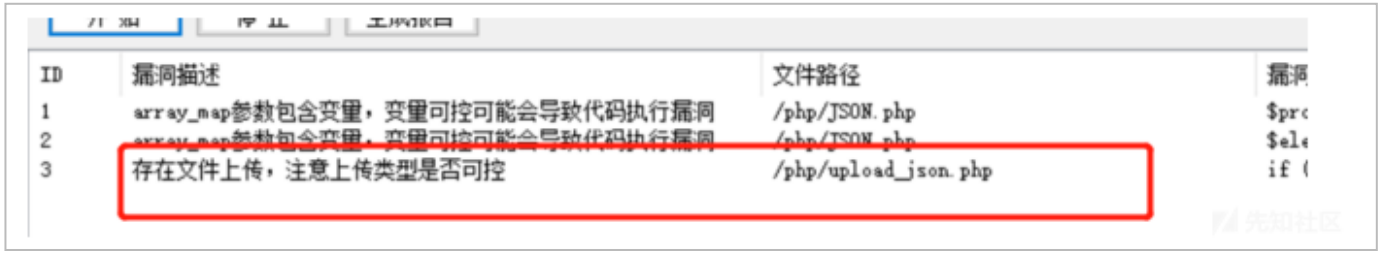
```php
$str = '0123456789';
$rand_str = '';
for ($i = 0; $i < 4; $i++){
    $k = mt_rand(1, strlen($str));
    $rand_str .= $str[$k - 1];
}
$_SESSION['verifycode'] = $rand_str;
imagefill($image, 0, 0, $bcolor);
imagestring($image, 7, 7, 10, $rand_str, $fcolor);
header('content-type:image/png');
imagepng($image);
?>
```


(https://xzfile.aliyuncs.com/media/upload/picture/20200429150937-62ef011c-89e8-1.png)

此外，值得一提的是，许多站点存在验证码重用漏洞所用的代码和使用方法都和本文的案例一样。

三、前台不能拿 shell 的上传漏洞

由于此 cms 默认存在的编辑器为 kindetior，并且 easy 代码审计显示存在一个文件上传



(https://xzfile.aliyuncs.com/media/upload/picture/20200429151333-efc59ee8-89e8-1.png)

查阅代码，发现存在可以直接上传 html 页面，txt 等页面

(https://xzfile.aliyuncs.com/media/upload/picture/20200429151350-f9fb94da-89e8-1.png)

更详细的可以查看 https://www.cnblogs.com/backlion/p/10421405.html，本文不再赘述。(https://www.cnblogs.com/backlion/p/10421405.html%EF%BC%8C%E6%9C%AC%E6%96%87%E4%B8%8D%E5%86%8D%E8%B5%98%E8%BF%B0%E3%80%82)

四、新手 SQL 注入采坑：

在 admin/login.php 后台登录的地方看到 a_name 和 a_password 没有经过处理就带入 SQL 语句中
以为存在后台万能密码

```php
<?php
require_once('../system/inc.php');
if(isset($_POST['submit'])){
    if ($_SESSION['verifycode'] != $_POST['verifycode']) {
        alert_href('验证码错误','cms_login.php');
    }
    null_back($_POST['a_name'],'请输入用户名');
    null_back($_POST['a_password'],'请输入密码');
    null_back($_POST['verifycode'],'请输入验证码');
    $a_name = $_POST['a_name'];
    $a_password = $_POST['a_password'];
    $sql = 'select * from xtcms_manager where m_name = "'.$a_name.'" and m_password = "'.md5($a_password)
        .'"';
    $result = mysql_query($sql);
    if(!! $row = mysql_fetch_array($result)){
        setcookie('admin_name',$row['m_name']);
        setcookie('admin_password',$row['m_password']);
        header('location:cms_welcome.php');
    }else{
        alert_href('用户名或密码错误','cms_login.php');
    }
}
```


(https://xzfile.aliyuncs.com/media/upload/picture/20200429151410-06085d26-89e9-1.png)

尝试万能密码 Payload 无果 头疼 ing！！

尝试 sqlmap 进行测试无果 头疼 ing！！！

最后看了一下 easy 代码审计系统里自带的 MySQL 监控发现对引号进行了转义

```
select * from xtcms_manager where m_name = "admin" and m_password = "e10adc3949ba59abbe56e057f20f883e"
SELECT COUNT(*) AS count FROM xtcms_user
SELECT COUNT(*) AS count FROM xtcms_user
SELECT COUNT(*) AS count FROM xtcms_vod
SELECT COUNT(*) AS count FROM xtcms_vod
SELECT COUNT(*) AS count FROM xtcms_user_pay
SELECT COUNT(*) AS count FROM xtcms_user_pay
SHOW VARIABLES
SHOW COLLATION
SET NAMES utf8;SET character_set_results=NULL
set names utf8
select * from xtcms_system where id = 1
set names utf8
select * from xtcms_system where id = 1
set names utf8
select * from xtcms_system where id = 1
select * from xtcms_manager where m_name = "\"admin\" or 1=1;#" and m_password = "e10adc3949ba59abbe56e057f20f883e"
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200429151431-124d1de2-89e9-1.png)

可是查看上面的代码又没有发现什么处理传参的地方，最后才在上方注意带包含了一个 inc.php

```php
require_once('../system/inc.php');
if(isset($_POST['submit'])){
    if ($_SESSION['verifycode'] != $_POST['verifycode'])
        alert_href('验证码错误','cms_login.php');
    }
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200429151444-1a1fb336-89e9-1.png)

inc.php 的内容

```php
<?php
require_once('conn.php');
require_once('library.php');
require_once('function.php');
require_once('config.php');
?>
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200429151455-20b349f6-89e9-1.png)

我们分别进入这四个文件进行审计，最后在 library.php 中发现

```php
if (!get_magic_quotes_gpc()) {
    if (!empty($_GET)) {
        $_GET = addslashes_deep($_GET);
    }
    if (!empty($_POST)) {
```

```
        $_POST = addslashes_deep($_POST);
    }
    $_COOKIE = addslashes_deep($_COOKIE);
    $_REQUEST = addslashes_deep($_REQUEST);
}
function addslashes_deep($_var_0)
{
    if (empty($_var_0)) {
        return $_var_0;
    } else {
        return is_array($_var_0) ? array_map('addslashes_deep', $_var_0) : addslashes($_var_0);
    }
}
```


(https://xzfile.aliyuncs.com/media/upload/picture/20200429151506-27698b48-89e9-1.png)

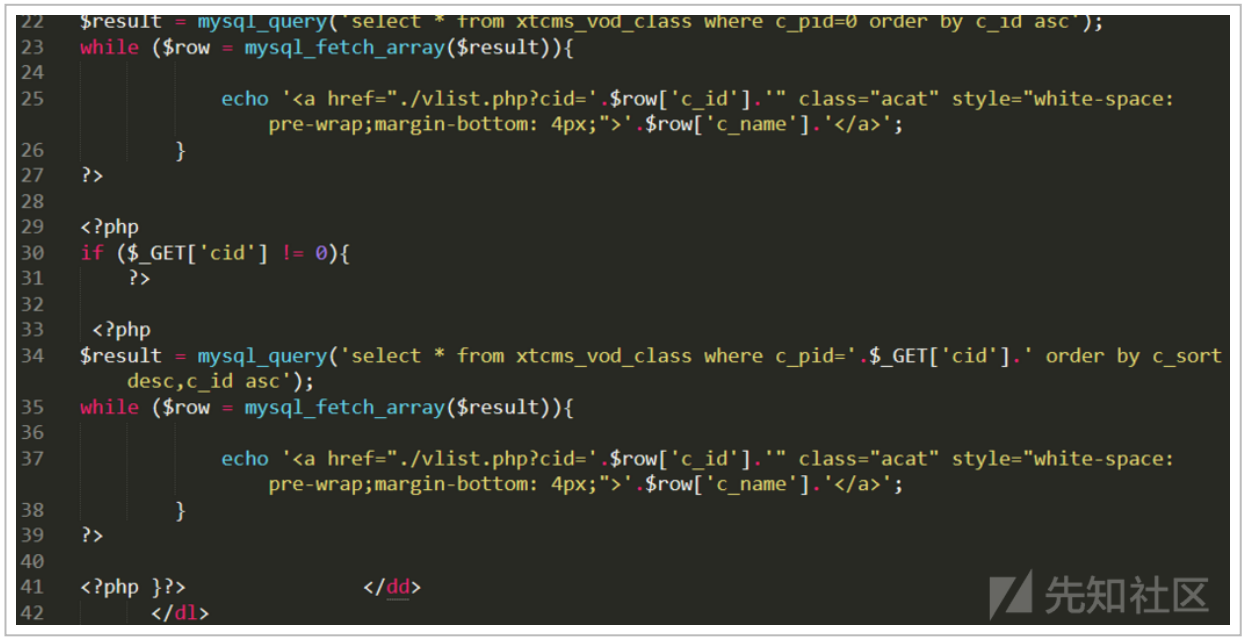
意思大概就是没有开启魔术引号的时候会将上述传参加上 addslashes，因此在这里的们的双引号自然就会被加上反斜杠了（addslashes 并不是万能的防护措施，具体可以去查阅百度）

五、前端 SQL 注入

输入一个引号发现回显不一样，审计此文件 /vlist.php

(https://xzfile.aliyuncs.com/media/upload/picture/20200429151537-39cd7416-89e9-1.png)

可以的清楚的看到没有任何防护措施，而由于传参方法为数字型，因此也没有使用双引号进行包裹，直接进入 sql 语句进行拼接



(https://xzfile.aliyuncs.com/media/upload/picture/20200429151552-42a62452-89e9-1.png)

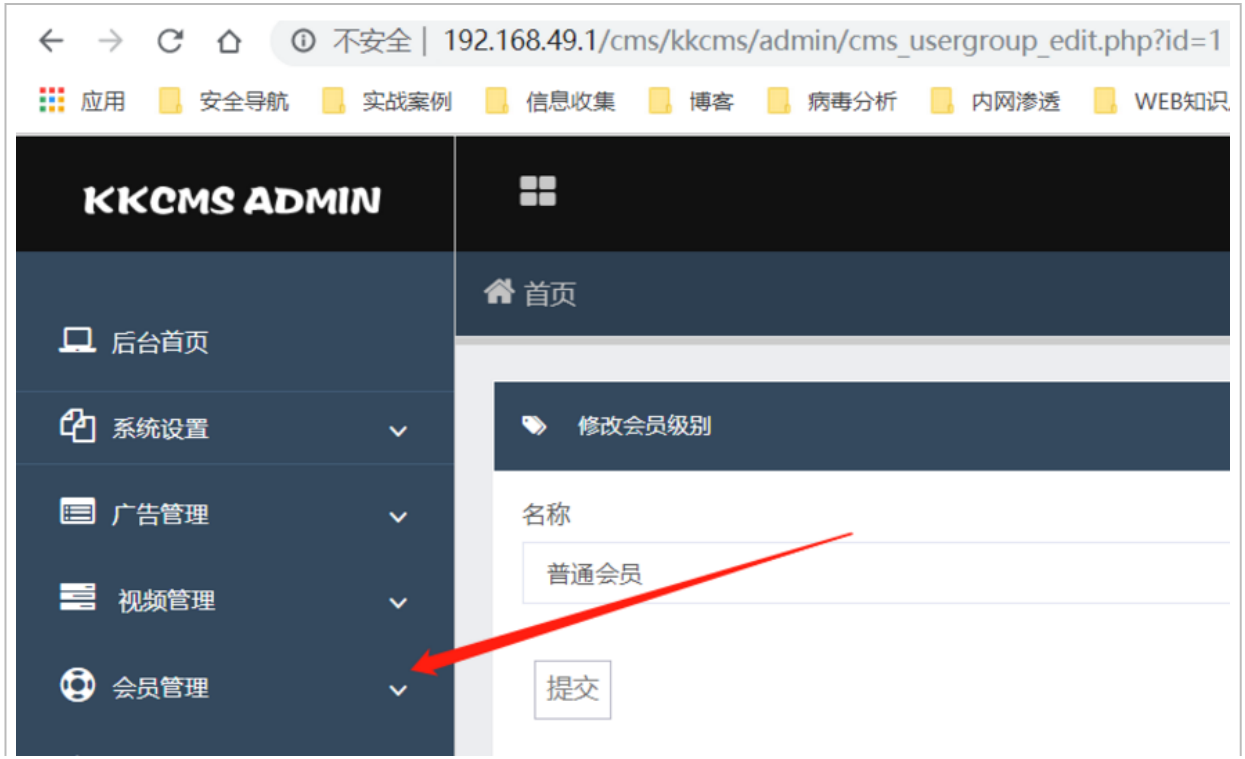
看到没有任何防护的措施直接上 Sqlmap
另外此处其实是有上了如采坑点所说的 addslashes 的，但是由于没有使用双引号将 $c_id 包裹起来，所以我们无需使用双引号进行闭合，addslashes 也就没有起到作用了。

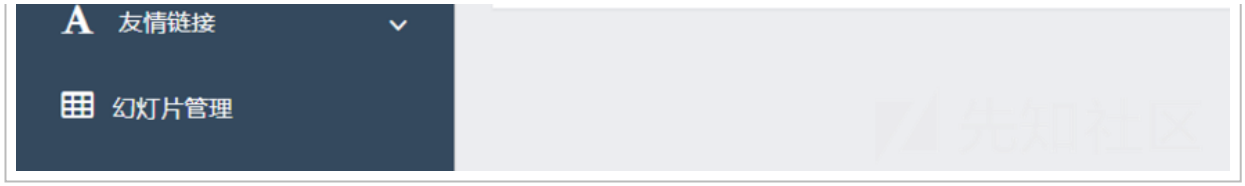
(这里手工注入的时候，不好进行判断，因为他默认 c_id 会等于 0，这会给我们的输入造成一定的的干扰)

```
$result = mysql_query('select * from xtcms_vod_class where c_pid=0 order by c_id asc');
```

六、后台 SQL 注入

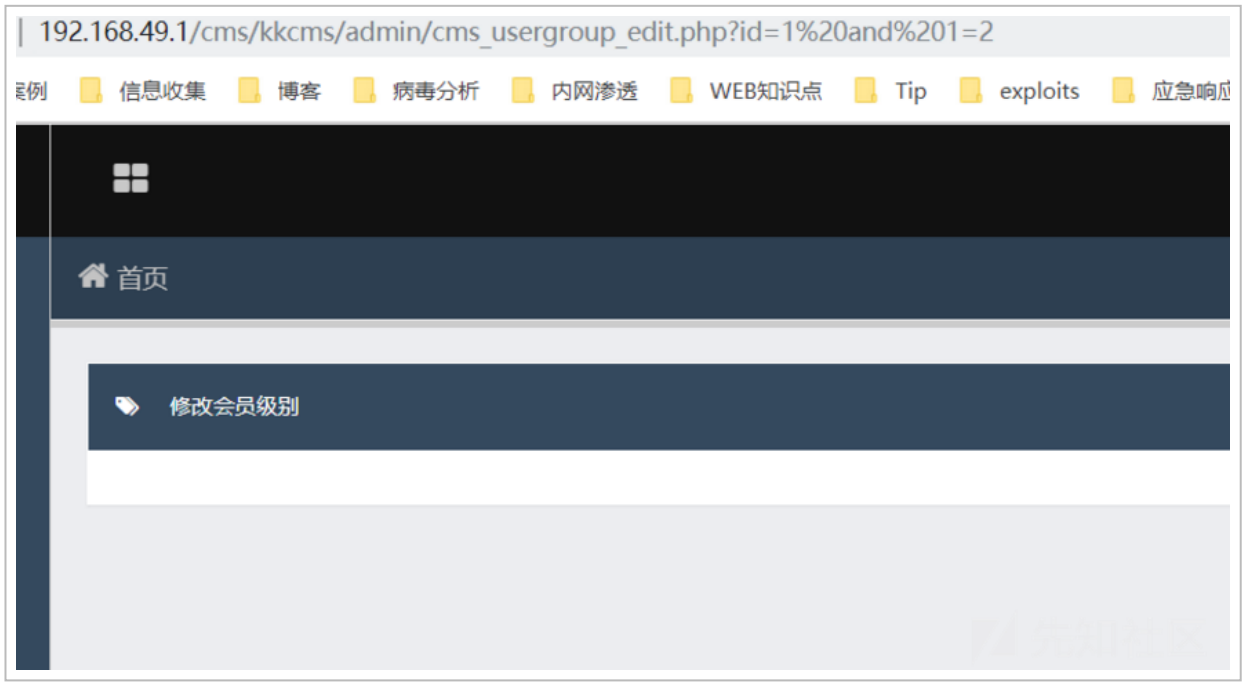
进入后台，随便点点，寻找与数据库交互的地方（尤其是传参数为数字型的参数，因为我们在前面已经发现字符型的参数使用双引号包裹，而存在的 addslashes 会阻止一切字符型 SQL 注入），发现在会员管理处存在 id=1





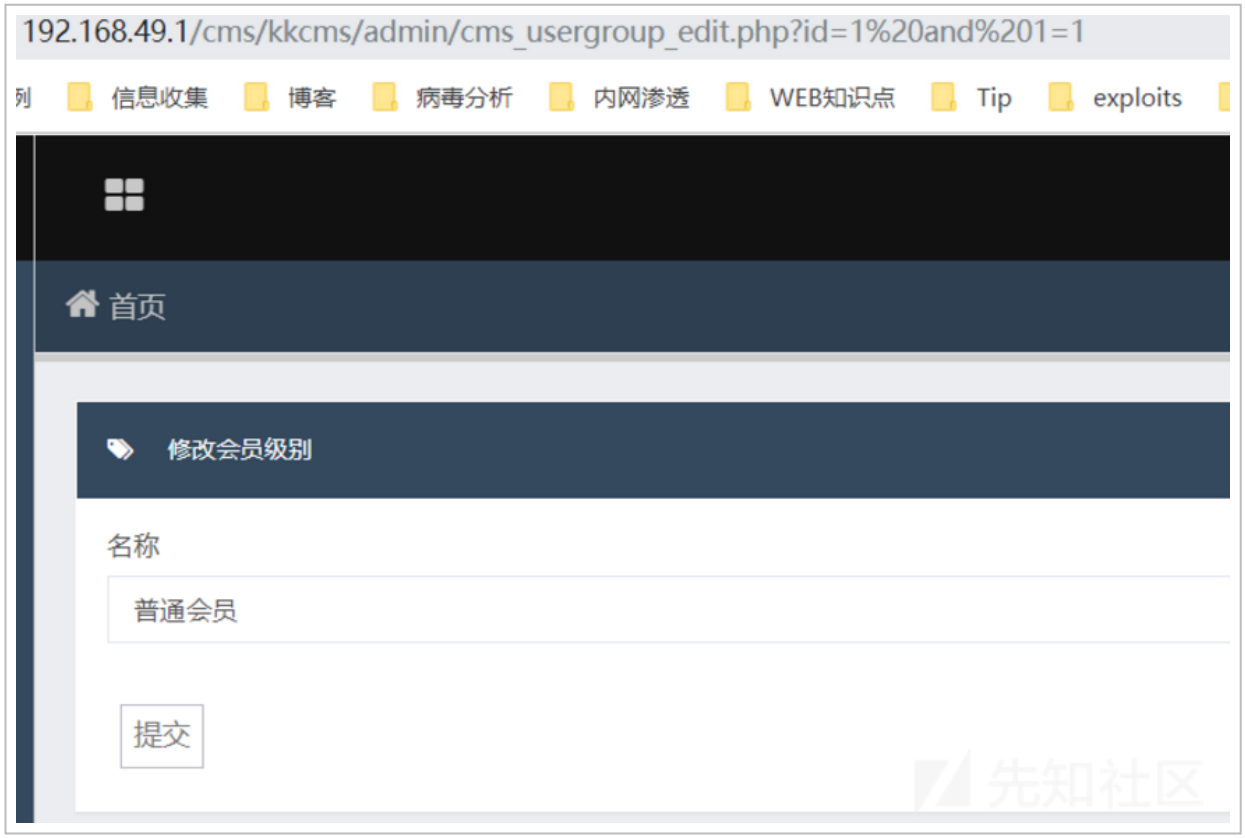
(https://xzfile.aliyuncs.com/media/upload/picture/20200429152346-5cce9c96-89ea-1.png)

id=1 and 1=2



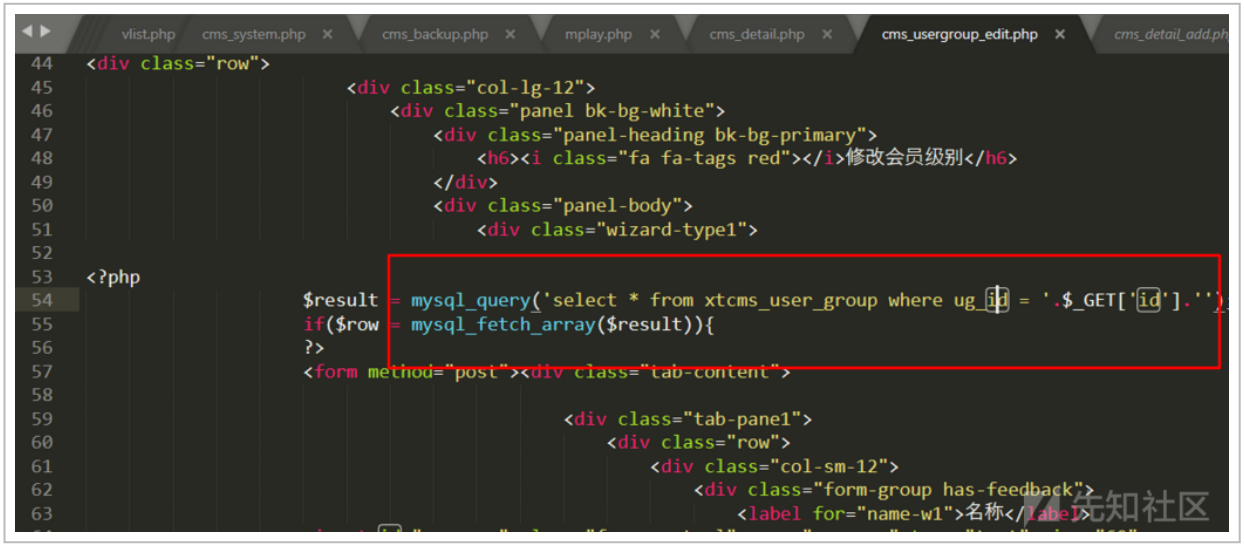
(https://xzfile.aliyuncs.com/media/upload/picture/20200429152355-628f121e-89ea-1.png)

id=1 and 1=1 一个非常明显的 sql 注入漏洞



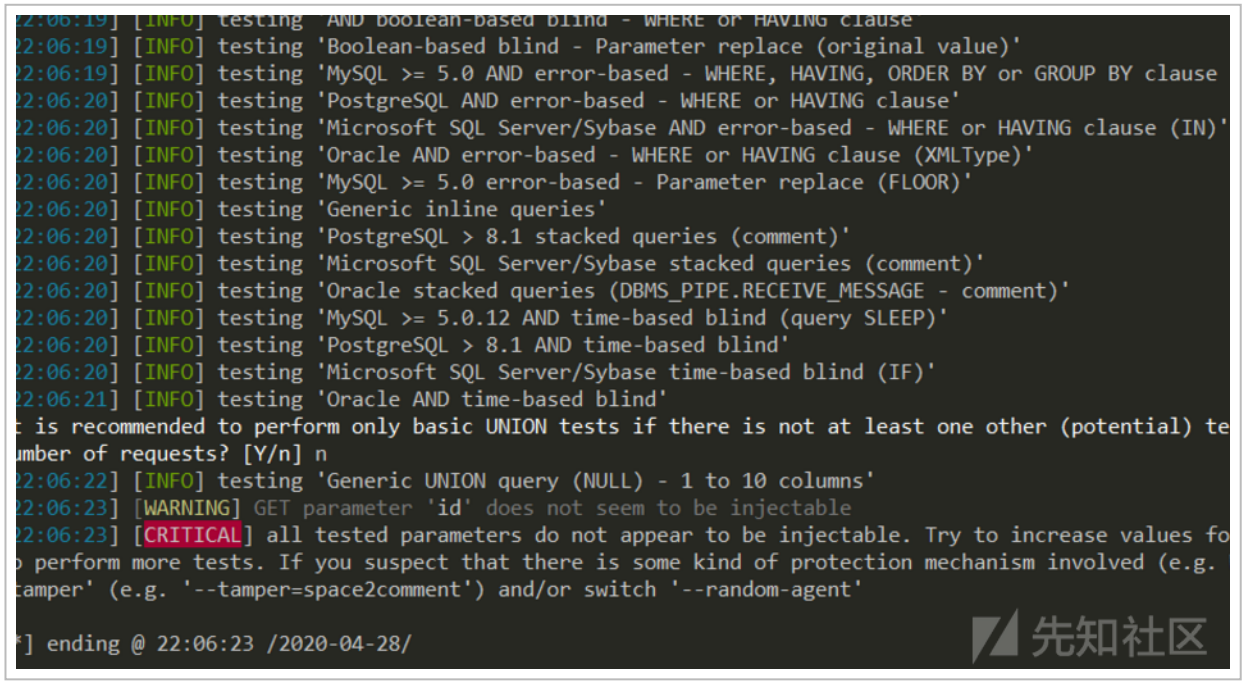
(https://xzfile.aliyuncs.com/media/upload/picture/20200429152405-6860f4aa-89ea-1.png)

进入文件中进行查看，直接进行拼接



(https://xzfile.aliyuncs.com/media/upload/picture/20200429152427-75687d30-89ea-1.png)

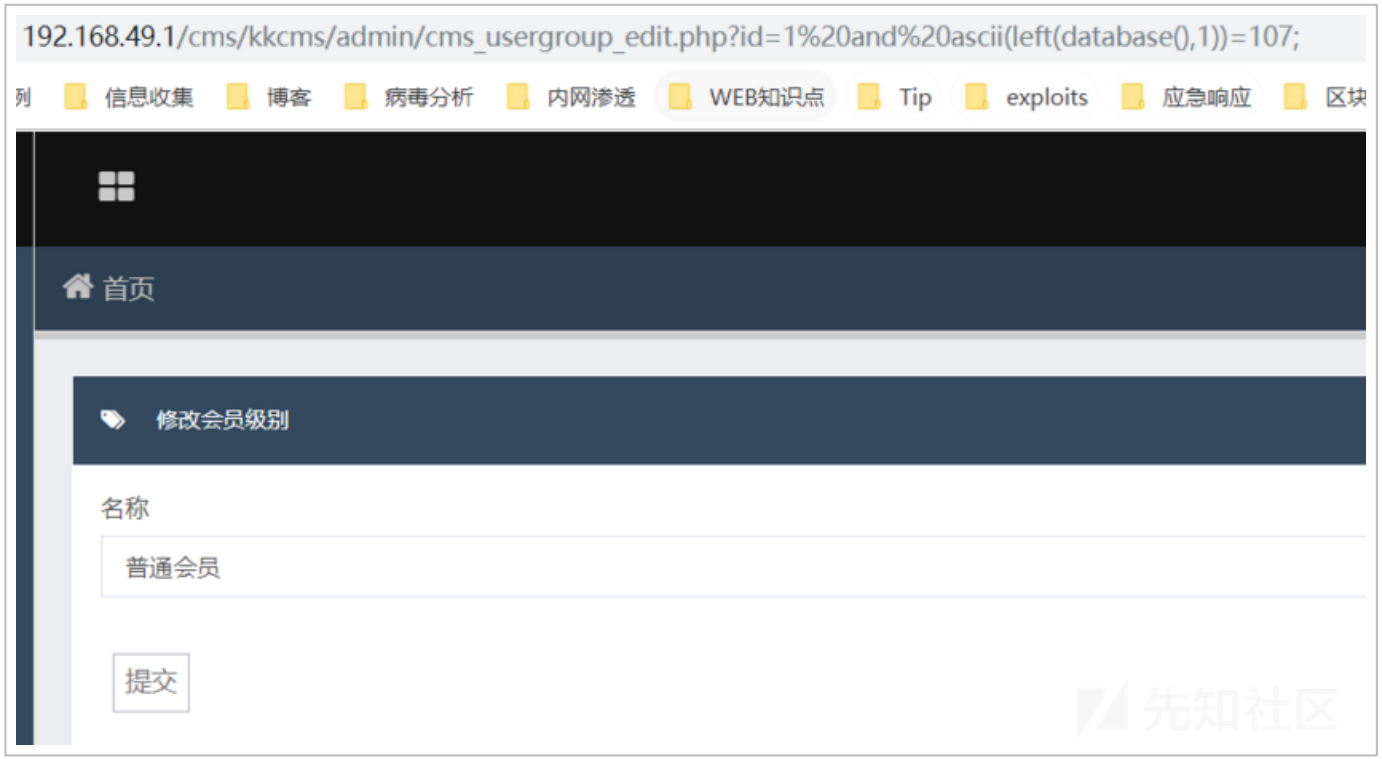
使用 sqlmap 进行注入，纳尼！这么明显的注入居然不存在？？？



(https://xzfile.aliyuncs.com/media/upload/picture/20200429152441-7e023b8e-89ea-1.png)

不过我们可以通过自己判断来确定存在一个盲注，确定当前数据库的第一个字母为 k

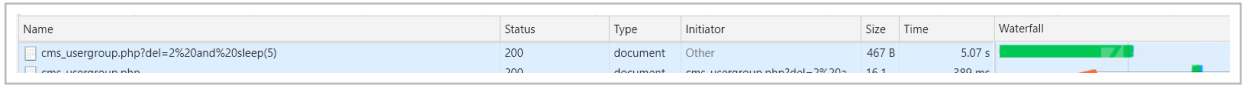
盲注代码不再赘述，这里仅做证明



(https://xzfile.aliyuncs.com/media/upload/picture/20200429152455-864e505c-89ea-1.png)

全后台 delete 注入

```
http://192.168.49.1/cms/kkcms/admin/cms_usergroup.php?del=2%20and%20sleep(5)
```



(https://xzfile.aliyuncs.com/media/upload/picture/20200429164753-1d0c3756-89f6-1.png)

而后台的代码很简单，直接引入数字型的 del 参数，带入数据库，整个后台的删除功能都是这样写的
存在于整个后台文件的共 12 个文件存在 delete 注入

```
2
3: if ($_GET['del'] == 1) {
4:  alert_back('默认账户不能删除！');
5: } else {
6:  if (isset($_GET['del'])) {
7:      $sql = 'delete from xtcms_manager where m_id= ' . $_GET['del'] . '';
8:      if (mysql_query($sql)) {
9:          alert_href('删除成功!', 'cms_admin.php');

E:\phpstudy\PHPTutorial\WWW\cms\kkcms\admin\model\adwei.php:
12:  }
13: }
14: if (isset($_GET['del'])) {
15:  $sql = 'delete from xtcms_adclass where id = ' . $_GET['del'] . '';
```

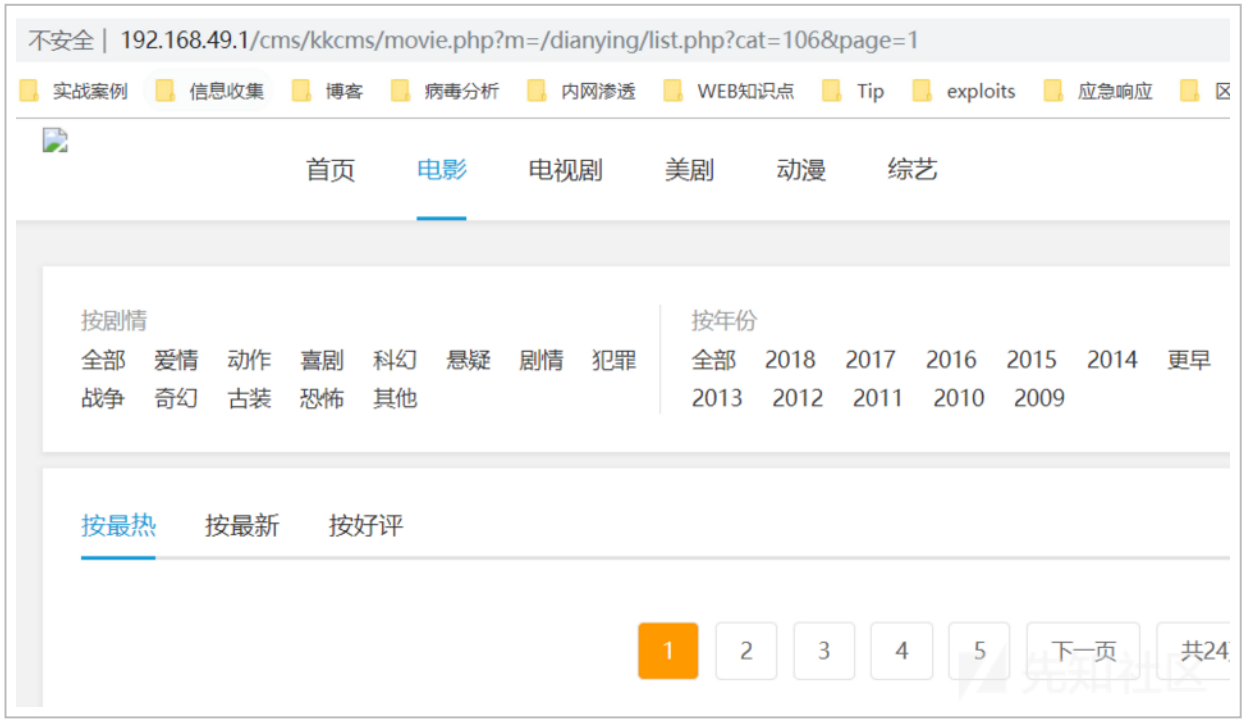

(https://xzfile.aliyuncs.com/media/upload/picture/20200429165022-7622026c-89f6-1.png)

七、多处 XSS 漏洞

1、随便点击一个页面，寻找存在的 get 请求的参数，这就是我们寻找 XSS 漏洞的突破点

其中已发现的存在以下同样用法的 XSS
template\wapian\movie.php
template\wapian\tv.php
template\wapian\zongyi.php
template\wapian\dongman.php

2、点击某处 URL 发现 URL 存在 m、cat 、page 等参数



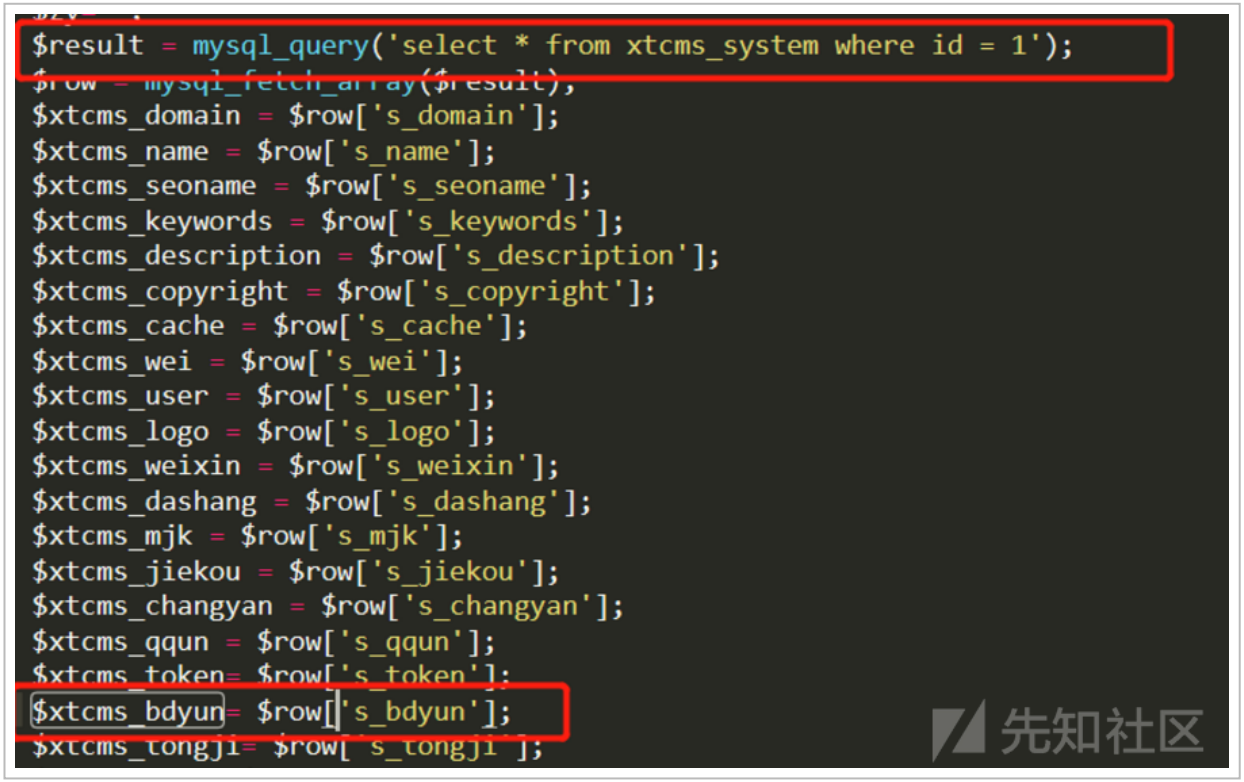
(https://xzfile.aliyuncs.com/media/upload/picture/20200429152529-9a791350-89ea-1.png)

3、我们先查看 movie.php 里的内容，发现包含了三个文件

```php
<?php
include 'system/inc.php';
include 'system/list.php';
$page=$_GET['page'];
include('template/'.$xtcms_bdyun.'/movie.php');?>
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200429152547-a50730ea-89ea-1.png)

4、跟进 include('template/'.$xtcms_bdyun.'/movie.php'); 中的 $xtcms_bdyun 是什么
全局搜索 $xtcms_bdyun 发现为数据库中 system 表的某个字段



(https://xzfile.aliyuncs.com/media/upload/picture/20200429152558-abd3732a-89ea-1.png)

```
mysql> select s_bdyun  from xtcms_system where id = 1
    -> ;
+---------+
| s_bdyun |
+---------+
| wapian  |
+---------+
```

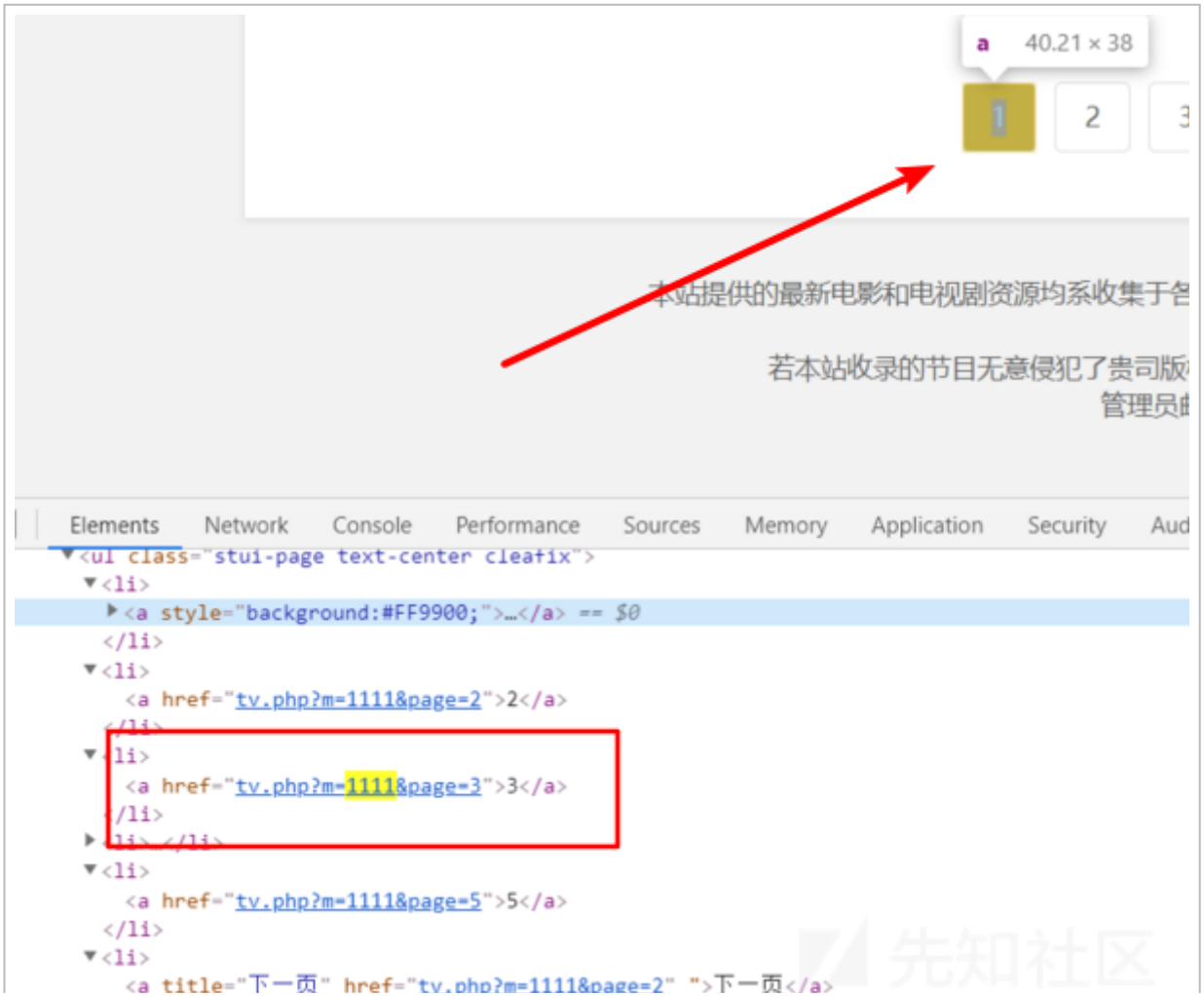
(https://xzfile.aliyuncs.com/media/upload/picture/20200429152607-b149bae4-89ea-1.png)

5、定位代码页面在 template/wapian/movie.php，存在变量 m，没有发现 echo 输出

```
<?php
$b = (strpos($_GET['m'], 'rank='));
$ye = substr($_GET['m'], $b + 5);
?>
<li <?php if ($ye == "rankhot") {
      echo 'class="active"';
    } elseif ($ye == "createtime" or $ye == "rankpoint") { } else {
      echo 'class="active"';
    }; ?>><a href="?m=/dianying/list.php?rank=rankhot&page=1">按最热</a></li>
<li <?php if ($ye == "createtime") {
      echo 'class="active"';
    } else { }; ?>><a href="?m=/dianying/list.php?rank=createtime&page=1">按
    </a></li>
<li <?php if ($ye == "rankpoint") {
      echo 'class="active"';
    } else { }; ?>><a href="?m=/dianying/list.php?rank=rankpoint&page=1">按好
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200429152630-beb76136-89ea-1.png)

6、只能根据页面关键字进行二次定位



(https://xzfile.aliyuncs.com/media/upload/picture/20200429152651-cb3c90de-89ea-1.png)

7、根据页面文字信息进行定位，终于发现了相关的 echo 信息

```
E:\phpstudy\PHPTutorial\WWW\cms\kkcms\template\wapian\dongman.php:
  141           <div class="hy-page clearfix">
  142             <ul class="cleafix">
  143:              <ul class="stui-page text-center cleafix">
  144                 <?php echo getPageHtml($page, $fenye, 'dongman.php?m=' . $yourne
  '&page='); ?><li><a>共<?php echo $fenye; ?>页</a></li>
  145               </ul>

E:\phpstudy\PHPTutorial\WWW\cms\kkcms\template\wapian\movie.php:
  168           </div>
  169           <div class="hy-page clearfix">
  170:            <ul class="stui-page text-center cleafix">
  171               <?php echo getPageHtml($page, $fenye, 'movie.php?m=' . $yourneed .
  '&page='); ?><li><a>共<?php echo $fenye; ?>页</a></li>
  172             </ul>

E:\phpstudy\PHPTutorial\WWW\cms\kkcms\template\wapian\tv.php:
  153           </div>
  154           <div class="hy-page clearfix">
  155:            <ul class="stui-page text-center cleafix">
  156               <?php echo getPageHtml($page, $fenye, 'tv.php?m=' . $yourneed . '&
  ?><li><a>共<?php echo $fenye; ?>页</a></li>
  157             </ul>

E:\phpstudy\PHPTutorial\WWW\cms\kkcms\template\wapian\zongyi.php:
  207           </div>
  208           <div class="hy-page clearfix">
  209:            <ul class="stui-page text-center cleafix">
  210               <?php echo getPageHtml($page, $fenye, 'zongyi.php?m=' . $yourneed
  '&page='); ?><li><a>共<?php echo $fenye; ?>页</a></li>
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200429152703-d23949e0-89ea-1.png)

8、根进 getpageHtml 函数 文件在 system/function.php，其实代码写的有点乱，看不大懂，但是我们发现它并没有对传参进行完整的过滤（上文 SQL 踩坑处的 addslashes 不足以过滤 xss），因此也就造成了 XSS 漏洞

```php
function getPageHtml($_var_60, $_var_61, $_var_62)
{
    $_var_63 = 5;
    $_var_60 = $_var_60 < 1 ? 1 : $_var_60;
    $_var_60 = $_var_60 > $_var_61 ? $_var_61 : $_var_60;
    $_var_61 = $_var_61 < $_var_60 ? $_var_60 : $_var_61;
    $_var_64 = $_var_60 - floor($_var_63 / 2);
    $_var_64 = $_var_64 < 1 ? 1 : $_var_64;
    $_var_65 = $_var_60 + floor($_var_63 / 2);
    $_var_65 = $_var_65 > $_var_61 ? $_var_61 : $_var_65;
    $_var_66 = $_var_65 - $_var_64 + 1;
    if ($_var_66 < $_var_63 && $_var_64 > 1) {
        $_var_64 = $_var_64 - ($_var_63 - $_var_66);
        $_var_64 = $_var_64 < 1 ? 1 : $_var_64;
        $_var_66 = $_var_65 - $_var_64 + 1;
    }
    if ($_var_66 < $_var_63 && $_var_65 < $_var_61) {
        $_var_65 = $_var_65 + ($_var_63 - $_var_66);
        $_var_65 = $_var_65 > $_var_61 ? $_var_61 : $_var_65;
    }
    if ($_var_60 > 1) {
        $_var_67 .= '<li><a  title="上一页" href="' . $_var_62 . ($_var_60 - 1) . '">上一页</a></li>';
    }
    for ($_var_68 = $_var_64; $_var_68 <= $_var_65; $_var_68++) {
        if ($_var_68 == $_var_60) {
            $_var_67 .= '<li><a style="background:#FF9900;"><font color="#fff">' . $_var_68 . '
                </font></a></li>';
        } else {
            $_var_67 .= '<li><a href="' . $_var_62 . $_var_68 . '">' . $_var_68 . '</a></li>';
        }
    }
    if ($_var_60 < $_var_65) {
        $_var_67 .= '<li><a  title="下一页" href="' . $_var_62 . ($_var_60 + 1) . '">下一页</a></li>';
```

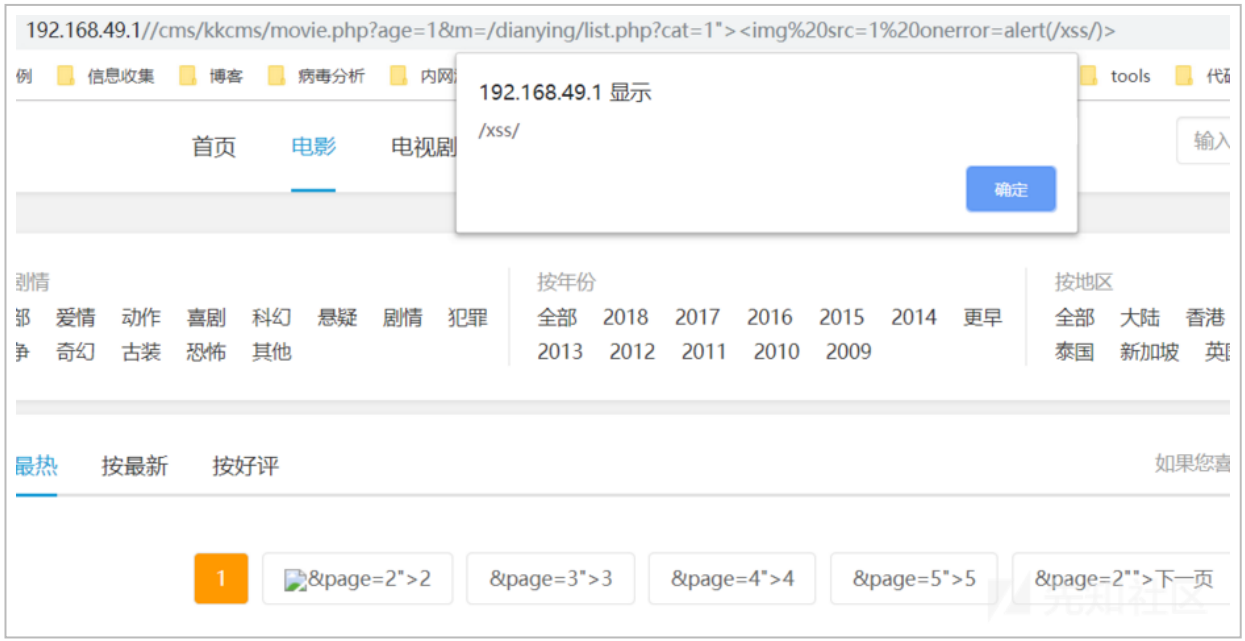
(https://xzfile.aliyuncs.com/media/upload/picture/20200429152720-dccd2976-89ea-1.png)

(https://xzfile.aliyuncs.com/media/upload/picture/20200429152728-e199101e-89ea-1.png)

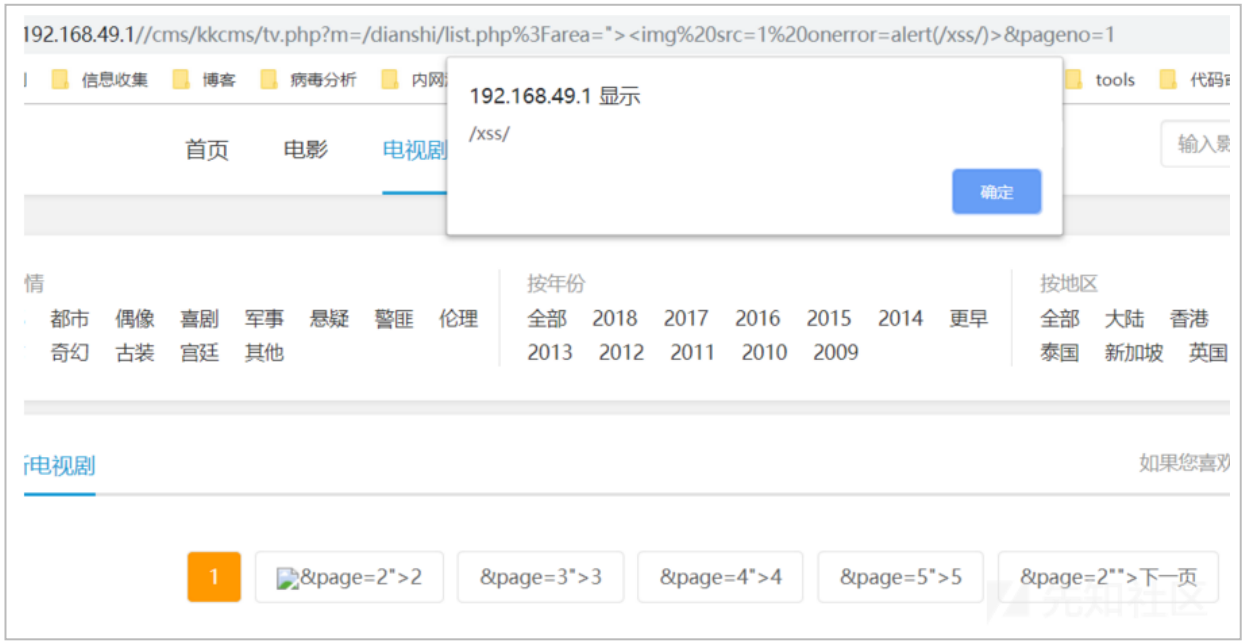
成功弹框

(https://xzfile.aliyuncs.com/media/upload/picture/20200429152745-ebbd29d6-89ea-1.png)

此外，另外存在几处 XSS，其原理也和上述 XSS 类似，输出点都在 page 点
例如: cat 参数 存在 XSS



(https://xzfile.aliyuncs.com/media/upload/picture/20200429152842-0d77be24-89eb-1.png)
area 参数存在 xss



(https://xzfile.aliyuncs.com/media/upload/picture/20200429152846-100a37c0-89eb-1.png)

八、留言板处前端 + 后端 XSS 双杀

在我决定弹反射型 XSS 弹到怀疑人生的时候，我陷入了沉思，我觉得应该要弹一个存储型 XSS 才够，因此我苦寻良久（easy 代码审计工具中没有将可能存在存储型 XSS 漏洞的这个点显示出来），最后终于在 book.php 找到了（在首页上乱点没点到）

打开页面是一个留言框

(https://xzfile.aliyuncs.com/media/upload/picture/20200429152959-3b8c1850-89eb-1.png)

对代码进行审计发现需要我们传入的内容需要存在汉字（\x7f-\xff）
如果不存在汉字，则弹出内容不合法
（此处的验证码无法显示，我们作弊一下跳过验证码验证，将验证码字段的！= 改为 ==）

```php
<?php
include('system/inc.php');
if(isset($_POST['submit'])){
    if ($_SESSION['verifycode'] == $_POST['verifycode']) {
        alert_href('验证码错误','book.php');
    }
    null_back($_POST['userid'],'请填写昵称');
    null_back($_POST['content'],'请填写留言内容');
    //判断昵称是否符合规则
    if (!preg_match("/[\x7f-\xff]+(?:\s*[\x7f-\xff]+)*/", $_POST['userid'])) {
        echo "<script>alert('昵称不合法!');location.href='book.php'</script>";
        die();
    }
    //判断留言内容是否符合规则
    //判断有一个空格以上但是又不能全部是空的字符串
    if (!ctype_space($_POST['content']) && !empty($_POST['content'])) {
        //匹配数字字母下划线空格中文 数量不能超过10个字
        if (!preg_match("/[\x7f-\xff]+(?:\s*[\x7f-\xff]+)*/", $_POST['content'])) {
            echo "<script>alert('内容填写不合法!');location.href='book.php'</script>";
            die;
        }
    } else {
        echo "<script>alert('内容填写不合法!');location.href='book.php'</script>";
        die;
    }
```


(https://xzfile.aliyuncs.com/media/upload/picture/20200429153034-504e85de-89eb-1.png)

通过正则的校验后，直接进入数据库，并且包含了 template 下的 book.php 文件

```php
    $data['userid'] = $_POST['userid'];
    $data['content'] =addslashes($_POST['content']);
    $data['time'] =date('y-m-d h:i:s',time());
    $str = arrtoinsert($data);
        $sql = 'insert into xtcms_book ('.$str[0].') values ('.$str[1].')';
    if(mysql_query($sql)){

alert_href('留言成功!小的马上为您准备相关资源！','book.php');
}
else{
alert_back('抱歉！服务器好像开小差了呢！');
    }


}
include('template/'.$xtcms_bdyun.'/book.php');
?>
```


(https://xzfile.aliyuncs.com/media/upload/picture/20200429153120-6bb37834-89eb-1.png)

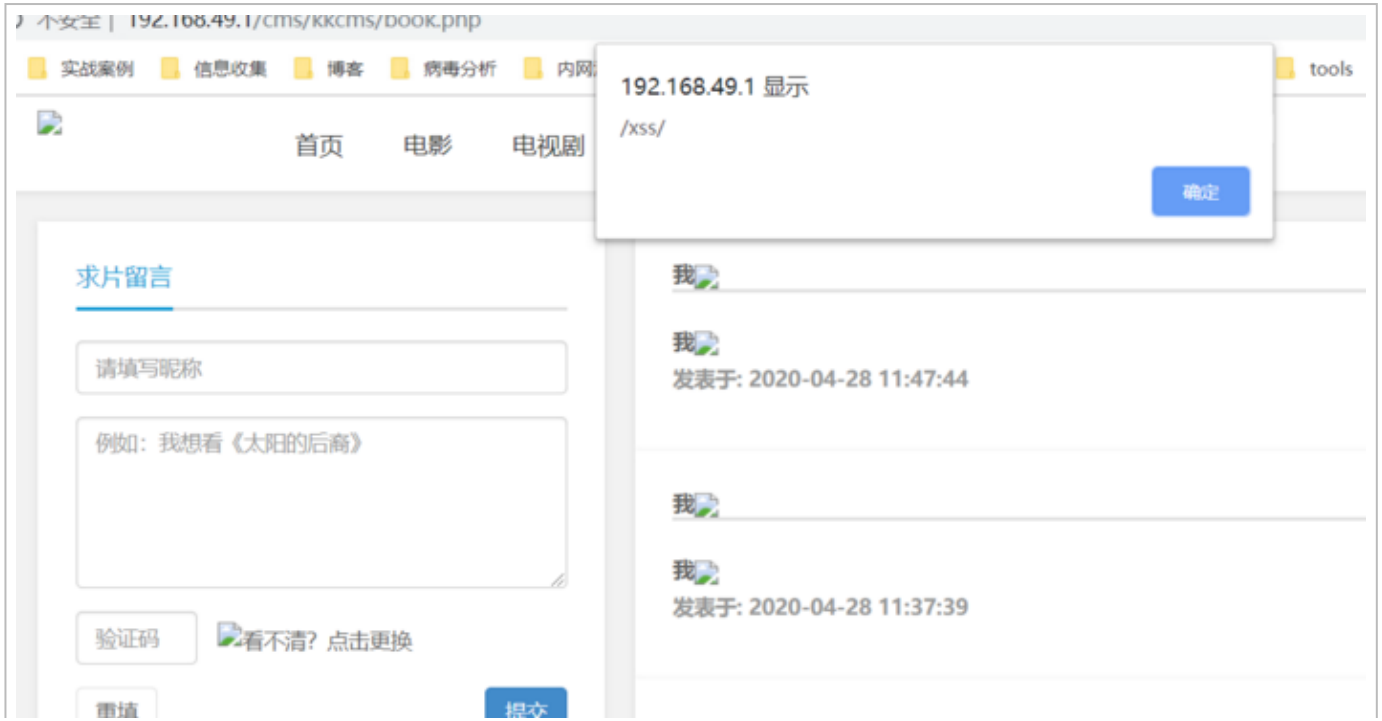
跟进此文件，发现从数据库中取出，然后直接 echo，产生 xss

```php
$sql = 'select * from xtcms_book order by id desc';
$pager = page_handle('page',10,mysql_num_rows(mysql_query($sql)));
```

```
53  $result = mysql_query($sql.' limit '.$pager[0].','.$pager[1].'');
54  while($row= mysql_fetch_array($result)){?>
55  <?php
56  $zhStr = $row['Reply'];
57  ?>
58  <div class="hy-layout clearfix" style="margin-top:0px">
59  <div class="hy-switch-tabs"><span class="text-muted pull-right">第<?php echo $row['id'] ?>楼</span>
60  <ul class="nav nav-tabs"><li><strong><?php echo $row['userid'] ?></strong></li></ul>
61  </div>
62  <div class="tab-content">
63  <div class="hy-play-list tab-pane fade in active" id="list3">
64  <div class="item">
65  <div class="plot">
66  <li><?php echo $row['content'] ?><br/>
67  <font color=red><?php if(0 < strlen($zhStr)){echo '管理员回复：';echo $row['Reply'];}?></font>
```

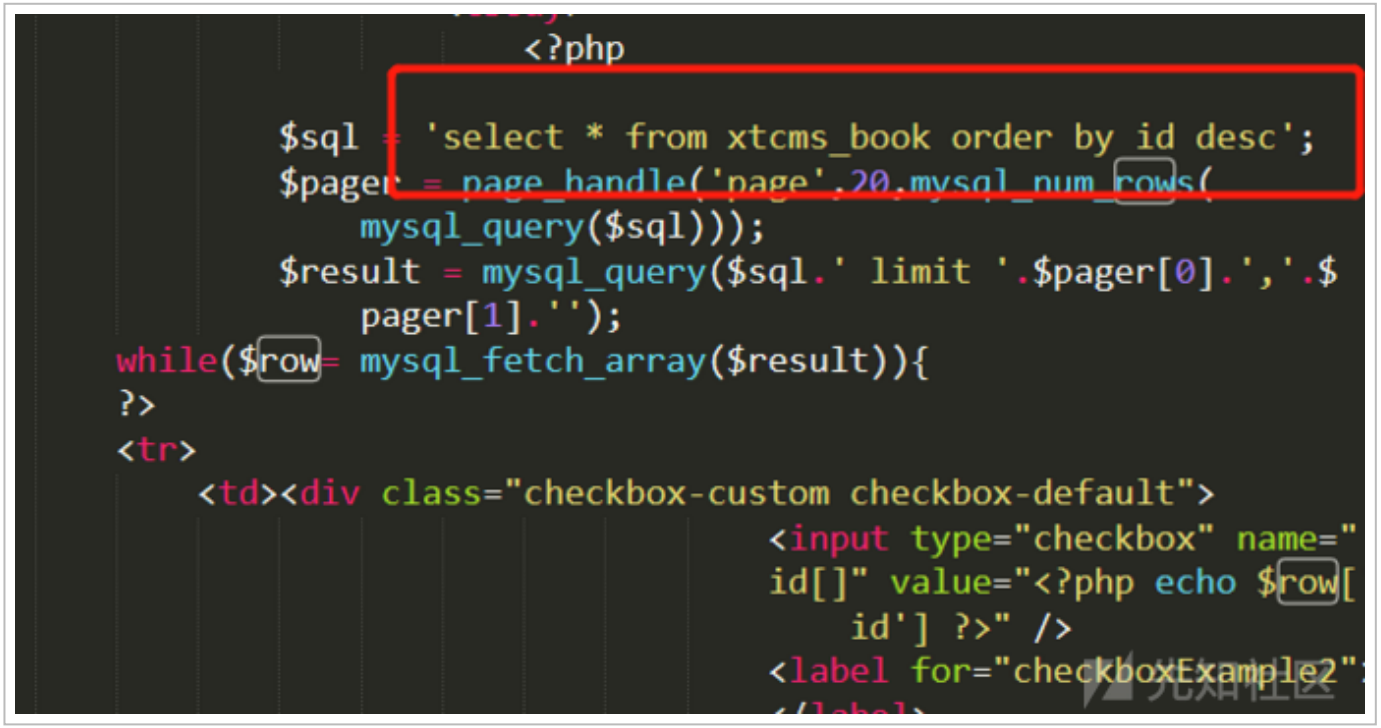
(https://xzfile.aliyuncs.com/media/upload/picture/20200429153545-098ebb90-89ec-1.png)

payload: 我 <img src=1 onerror=alert(/xss/)>



(https://xzfile.aliyuncs.com/media/upload/picture/20200429153615-1bbd6b54-89ec-1.png)

后台的留言板块文件为 cms_book.php

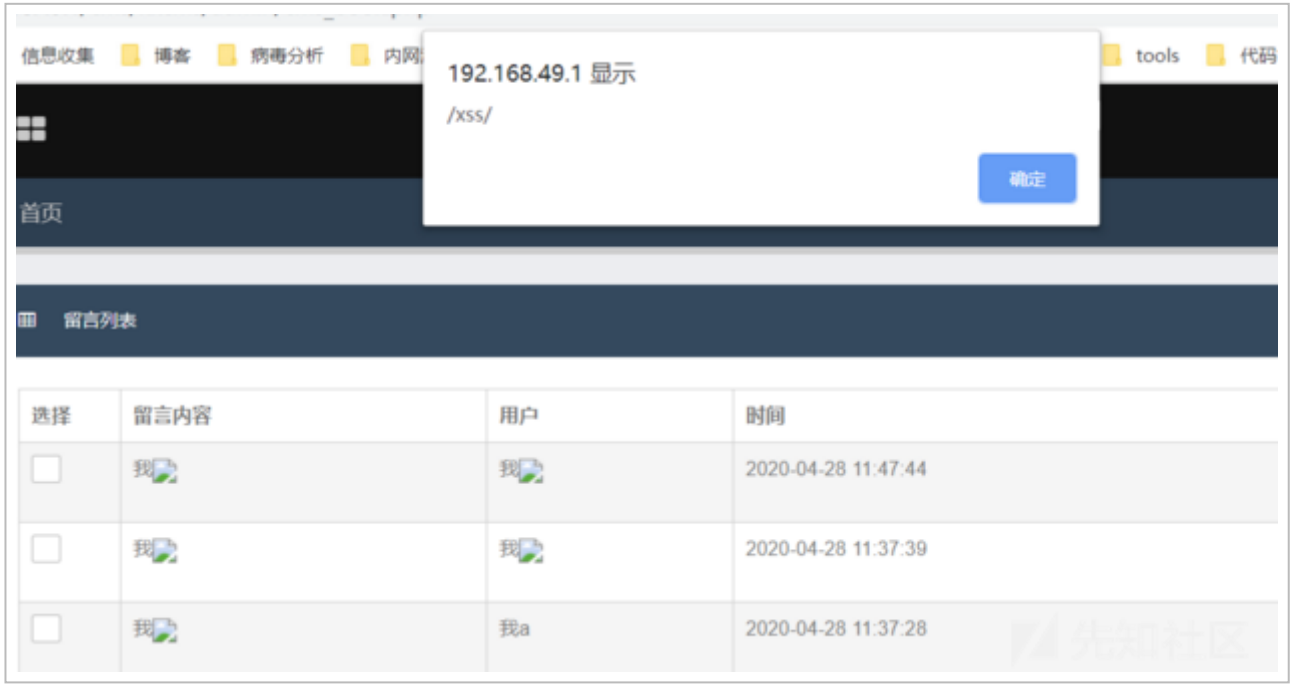


(https://xzfile.aliyuncs.com/media/upload/picture/20200429153659-35fa4b7c-89ec-1.png)

同样从数据库中直接取出数据，因此造成后台存储型 XSS

id	content	userid	Reply	time
6	鎴?img src=1 onerror=alert(/xss/)>	鎴?img src=1	NULL	2020-04-28 11:47:44
5	鎴?img src=1 onerror=alert(/xss/)>	鎴?img src=1	NULL	2020-04-28 11:37:39
4	鎴?img src=1 onerror>	鎴慳	NULL	2020-04-28 11:37:28

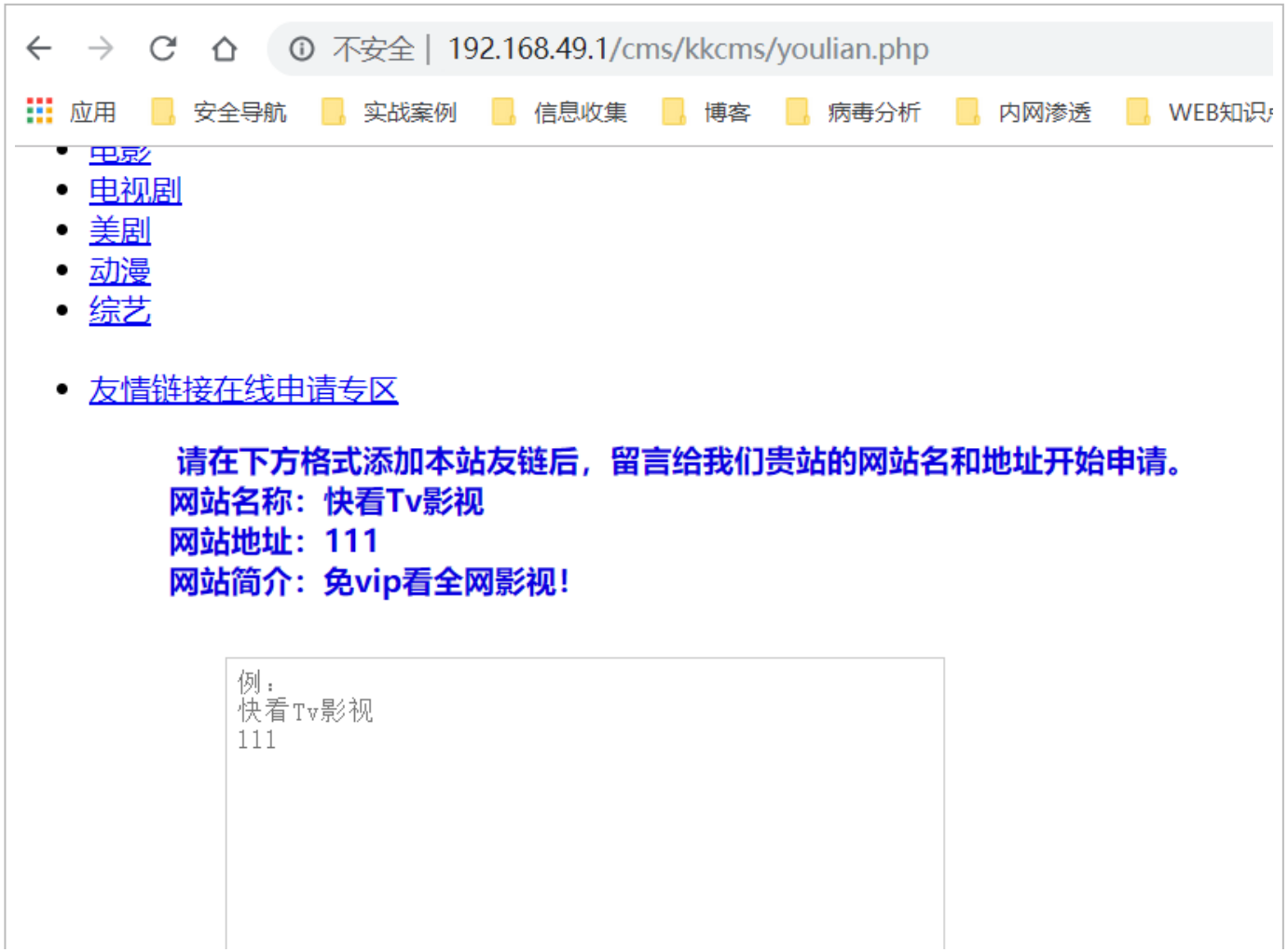
(https://xzfile.aliyuncs.com/media/upload/picture/20200429153719-41780822-89ec-1.png)



(https://xzfile.aliyuncs.com/media/upload/picture/20200429153728-474c7814-89ec-1.png)

九、第二个存储型 XSS

在 youlian.php 下存在可以提交友情链接申请的功能



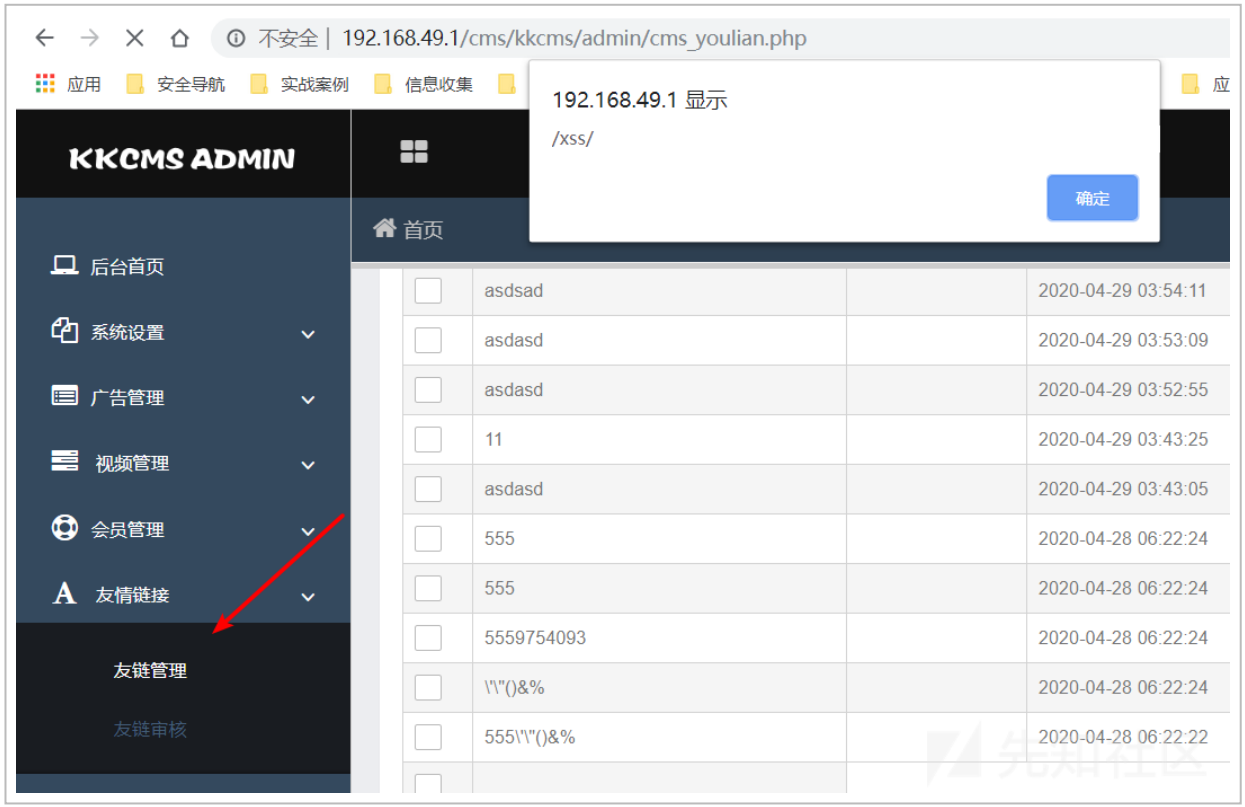
(https://xzfile.aliyuncs.com/media/upload/picture/20200429154530-6642a2c4-89ed-1.png)

同样是经过一个 addslsashes 的过滤然后传入数据库中，这同样无法阻止 XSS

```php
<?php
include('system/inc.php');
if(isset($_POST['submit'])){
null_back('admin','.');
    null_back($_POST['content'],'你的链接及网站名');
    $data['userid'] = $_POST['userid'];
    $data['content'] =addslashes($_POST['content']);
    $data['time'] =date('y-m-d h:i:s',time());

    $str = arrtoinsert($data);
        $sql = 'insert into xtcms_youlian ('.$str[0].') values ('.$str[1].')';
    if(mysql_query($sql)){

alert_href('申请成功！请耐心等待管理员核实！谢谢！','youlian.php');
}
else{
alert_back('申请失败！请联系网站底部邮箱申请吧！');
    }


}
include('template/'.$xtcms_bdyun.'/youlian.php');
?>
```

(https://xzfile.aliyuncs.com/media/upload/picture/20200429155902-4a8dc732-89ef-1.png)

传入后端 友链管理处，触发存储型 XS

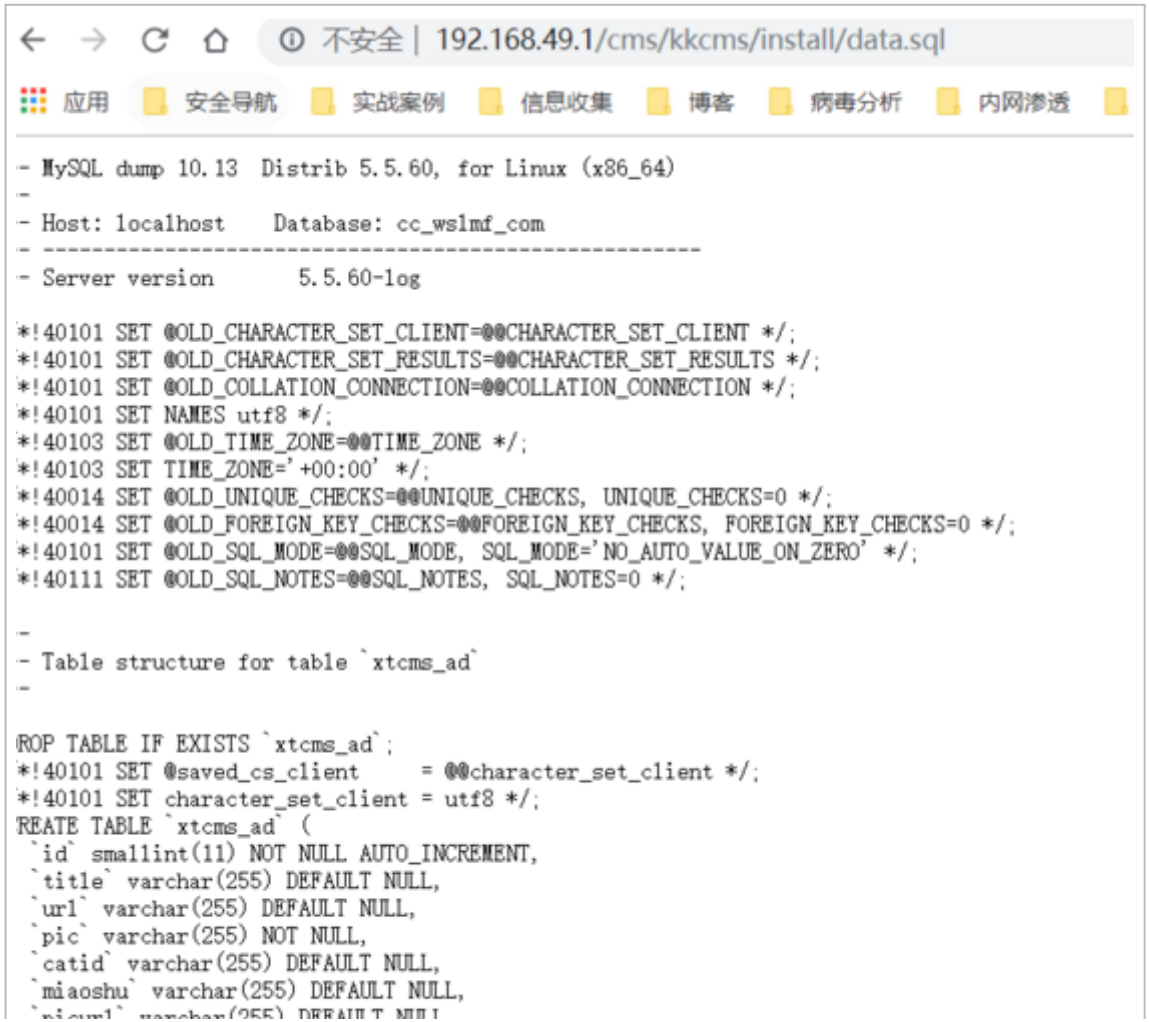


(https://xzfile.aliyuncs.com/media/upload/picture/20200429160021-794bd762-89ef-1.png)

十、数据库信息泄露

本来是要测试是否存在重装漏洞的，却发现存在一个 data.sql

打开网页发现存储的是默认创建的表结构和数据库信息，这是属于一个相对严重的信息泄露漏洞



```
 `picurl` varchar(255) DEFAULT NULL,
 `link` varchar(255) DEFAULT NULL,
 PRIMARY KEY (`id`)
 ENGINE=MyISAM AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;
*!40101 SET character_set_client = @saved_cs_client */;
```



(https://xzfile.aliyuncs.com/media/upload/picture/20200429153747-529a20b8-89ec-1.png)

十一、总结：

- 1：新手在入门审计 CMS 的时候，要有思路的去寻找漏洞点，不要看到一个点觉得存在，然后就跳跃性 - 的转移文件审计代码，这样会导致效率很低。
- 2：在审计 CMS 的时候，多注意一下是否默认存在开源软件或编辑器。
- 3：seay 代码审计工具里的 MYSQL 查询语句监控真的非常好用！
- 4：注意一些非 HTML 或者 PHP 后缀的文件，可能存在信息泄露漏洞。
- 5：审计 CMS 时，可以先从重装文件入手，因为这是整个 CMS 的开端。
- 6：一个十分寻常的 XSS 背后可能经过了多层跟进，代码审计的魅力就在于此。
- 7：本文比较简单，还望各位大佬多加海涵，也希望能够帮助一些刚入门代码审计的表哥们！