

# emlog CMS 的代码审计: 越权到后台 getshell - 先知社区

“ 先知社区, 先知安全技术社区 ”

## 前言

学习 CTF 这么久还没真正意义上审计过一款 cms, 这次决定花点时间去审计一款 cms 作为代码审计提升的跳板。由于相关要求, 这里就省去 cms 全名了, 主要分享一下学习思路。

## 代码审计

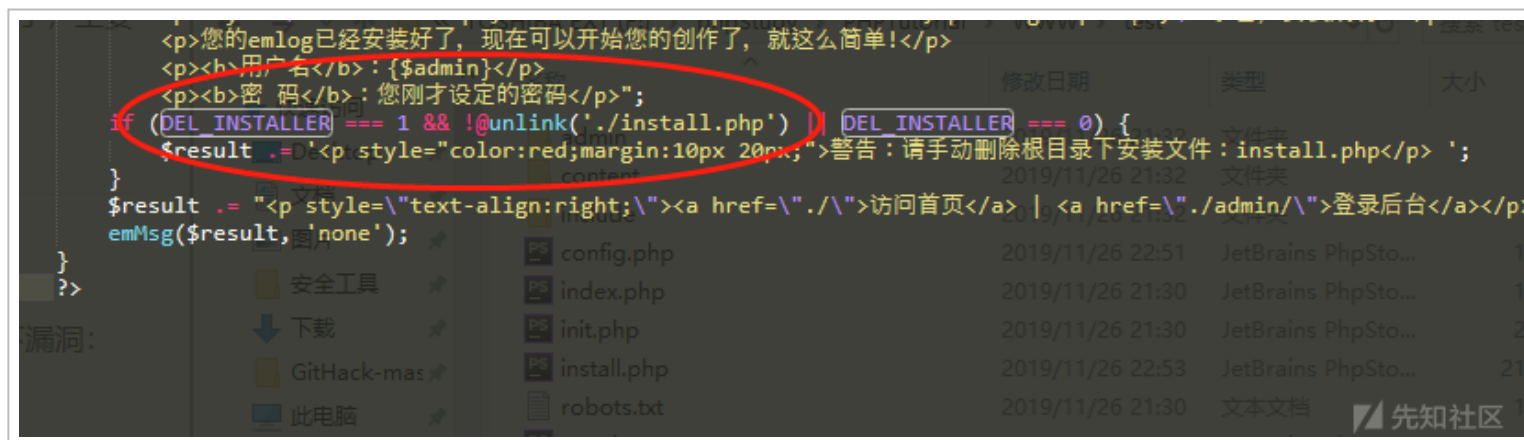
### 安装漏洞

其实一般代码审计都是从安装文件开始审计, 一般安装脚本主要存在如下漏洞:

- 无验证功能, 任意重装覆盖
- 表单不做过滤写入 config.php 导致 getshell
- \$\_GET['step'] 跳过限制步骤

漏洞文件: `install.php`

首先我们直奔第一个点能否任意重装, 我们可以看到必须常量 `DEL_INSTALLER` 为 1 的时候才会触发删除 `install.php`, 那么我们追踪 `DEL_INSTALLER` 看看



(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221220-ed09d88c-111f-1.png>)

这里可以看到 `DEL_INSTALLER` 默认值就是 0, 所以一般情况下这里是任意重装的, 我们从黑盒的测试也可以印证这一点。同时表单也做了过滤所以这里也没有后面两种情况。

```
define('EMLOG_ROOT', dirname(__FILE__));
define('DEL_INSTALLER', 0);
require_once EMLOG_ROOT . '/include/lib/function
```

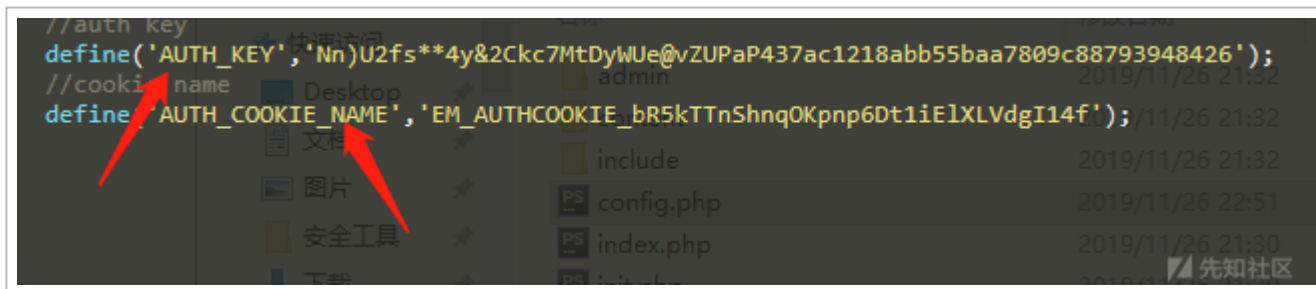
```
header('Content-Type: text/html; charset=UTF-8');
spl_autoload_register("emAutoload");
doStripslashes();

if(isset($_GET['action']) && $_GET['action'] != 'login') {
    $act = isset($_GET['action']) ? $_GET['action'] : 'index';
}
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221244-fb7b423e-111f-1.png>)

## 越权漏洞

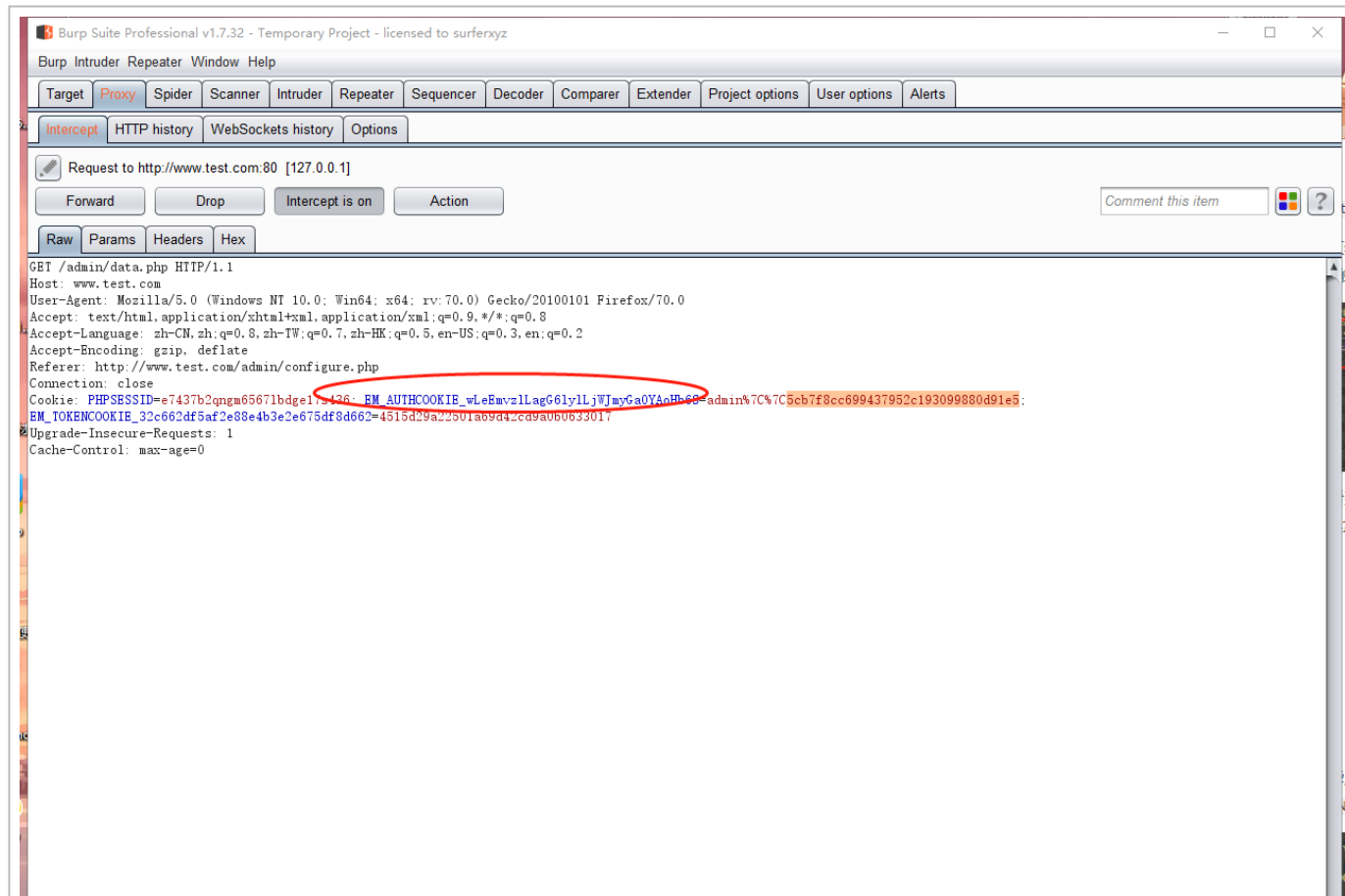
在安装完毕后打开 `config.php` 看到两个比较奇怪的常量定义：`AUTH_KEY` 和 `AUTH_COOKIE_NAME` 从名字来看这连个常量肯定是有一定联系的。如下图：



```
//auth key
define('AUTH_KEY', 'Nn)U2fs**4y&2Ckc7MtDyWUe@vZUPaP437ac1218abb55baa7809c88793948426');
//cookie name
define('AUTH_COOKIE_NAME', 'EM_AUTHCOOKIE_bR5kTTnShnqOKpnp6Dt1iElXLVdgI14f');
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221307-09002c76-1120-1.png>)

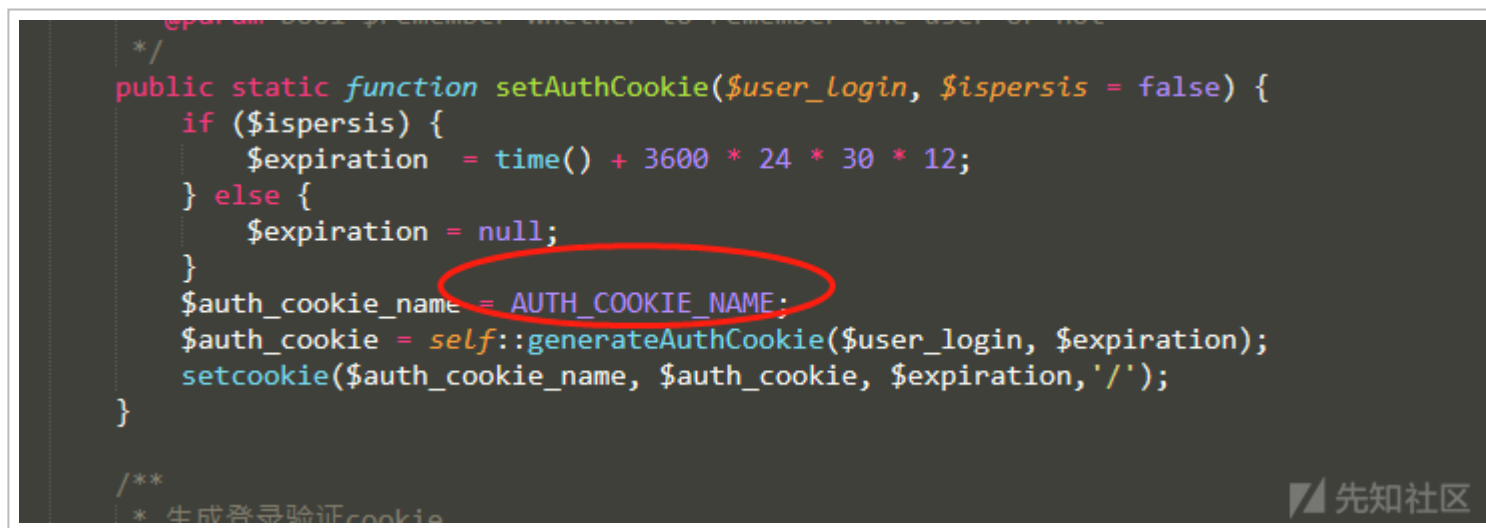
我们在抓包时候发现了 `AUTH_COOKIE_NAME` 这个常量，说明这是一个 cookie 名。我们继续在代码中追踪这个常量。





(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221336-1a81ee30-1120-1.png>)

我们可以看到在登录验证 cookie 中使用了这个常量，我们追踪一下 cookie 值是如何构造的



(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221352-240827a8-1120-1.png>)

这里可以看到，调用了 `emHash` 这个类方法，并且这里使用了 `AUTH_KEY` 这个常量，这里说明这两个常量是有联系的。我们在继续追踪 `hash_hmac()` 到底使用 `key` 做了什么。

```

    * @param int $user_id user login
    * @param int $expiration Cookie expiration in seconds
    * @return string Authentication cookie contents
    */
    private static function generateAuthCookie($user_login, $expiration) {
        $key = self::emHash($user_login . '|' . $expiration);
        $hash = hash_hmac('md5', $user_login . '|' . $expiration, $key);

        $cookie = $user_login . '|' . $expiration . '|' . $hash;

        return $cookie;
    }

    /**
     * Get hash of given string.
     *
     * @param string $data Plain text to hash
     * @return string Hash of $data
     */
    private static function emHash($data) {
        $key = AUTH_KEY;
        return hash_hmac('md5', $data, $key);
    }

    /**

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221408-2d5dfbf2-1120-1.png>)

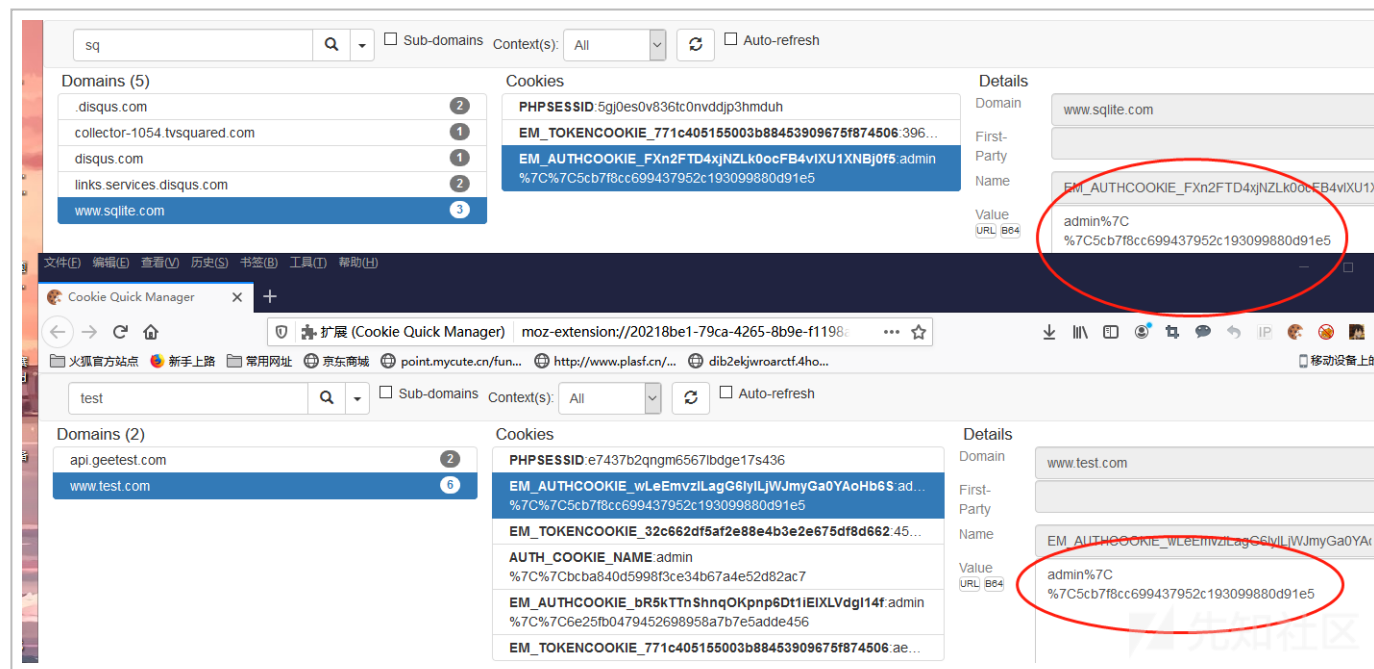
这里对传入的 `$key` 也就是 `AUTH_KEY` 进行了 md5 加密到一个二进制字符串中而后分割为 64 个字节与一个字符 \* 64 次的字符串进行异或最后得到两个字符串 `$ipad` , `$opad` 最后再将他们打包拼接用 md5 加密返回给上级调用。我们再回到上级。

```
if(!function_exists('hash_hmac')) {  
    function hash_hmac($algo, $data, $key) {  
        $packs = array('md5' => 'H32', 'sha1' => 'H40');  
  
        if (!isset($packs[$algo])) {  
            return false;  
        }  
  
        $pack = $packs[$algo];  
  
        if (strlen($key) > 64) {  
            $key = pack($pack, $algo($key));  
        } elseif (strlen($key) < 64) {  
            $key = str_pad($key, 64, chr(0));  
        }  
  
        $ipad = (substr($key, 0, 64) ^ str_repeat(chr(0x36), 64));  
        $opad = (substr($key, 0, 64) ^ str_repeat(chr(0x5C), 64));  
  
        return $algo($opad . pack($pack, $algo($ipad . $data)));  
    }  
}
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221449-45b47b86-1120-1.png>)

我们知道 `generateAuthCookie` 方法中的 `$key`、`$hash` 是由 `AUTH_KEY` 加密而成。最终的 cookie 是由 `$user_login`、`$expiration` `$hash` 拼接而成，而 `$expiration` 是 cookie 的生存时间，`$user_login` 是用户名。这里可以得知 `$key`、`$user_login`、`$expiration` 都是固定的那么只要知道 `AUTH_KEY` 就有伪造 cookie 造成越权的可能。

例如我们准备两个靶机，一个靶机登陆，获取这个靶机的 cookie 即可越权登陆另外一个靶机。前提是两个靶机的 `AUTH_KEY` 得一致。这里有点鸡肋但是还是有利用的可能。我们将两个靶机的 cookie 拿出来比较确实是一样的。

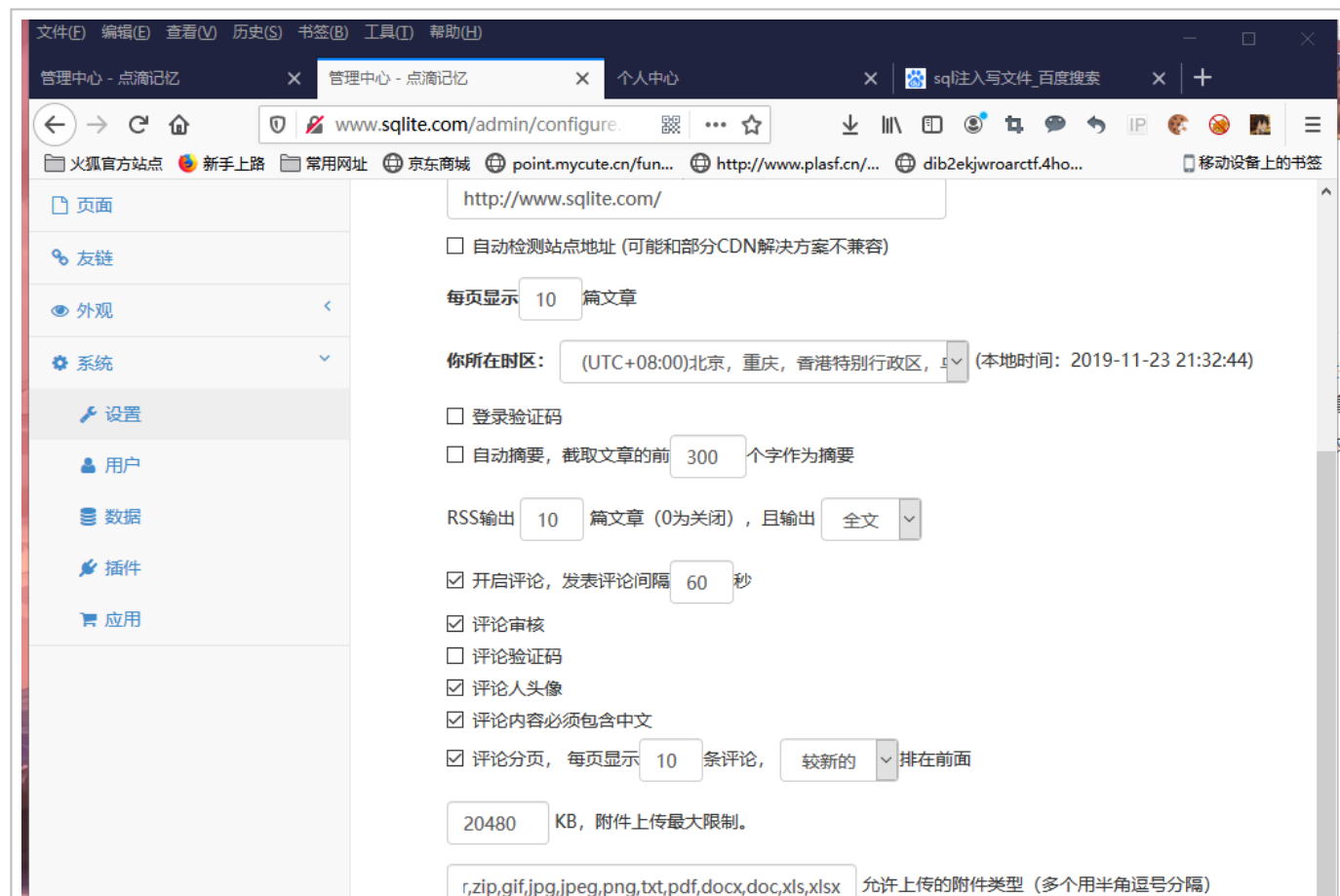


(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221512-53467e48-1120-1.png>)



## 后台 getsnelli (一)

其实上述两个漏洞已经能够让我们进入后台了，现在的任务就是如何 Getshell 了。这里我随便看了一下发现后台有设置上传附件后缀的功能。



上传图片生成缩略图, 最大尺寸: 420 x 460 (单位: 像素)

ICP备案号:

首页底部信息(支持html, 可用于添加流量统计代码):

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221655-9095be94-1120-1.png>)

但是测试发现, 加入 php 后缀你会发现将 phpt 替换为 X, 从源码中可有很直观看到这点。

```
163 comment_needchinese => isset($_POST['comment_needchinese']) ? addslashes($_POST['comment_needchinese']) : 'n',
164 'comment_interval' => isset($_POST['comment_interval']) ? intval($_POST['comment_interval']) : 15,
165 'iscomment' => isset($_POST['iscomment']) ? addslashes($_POST['iscomment']) : 'n',
166 'ischkcomment' => isset($_POST['ischkcomment']) ? addslashes($_POST['ischkcomment']) : 'n',
167 'isexcerpt' => isset($_POST['isexcerpt']) ? addslashes($_POST['isexcerpt']) : 'n',
168 'excerpt_subnum' => isset($_POST['excerpt_subnum']) ? intval($_POST['excerpt_subnum']) : 300, 多个用半角逗号分隔
169 'isthumbnail' => isset($_POST['isthumbnail']) ? addslashes($_POST['isthumbnail']) : 'n',
170 'rss_output_num' => isset($_POST['rss_output_num']) ? intval($_POST['rss_output_num']) : 10,
171 'rss_output_fulltext' => isset($_POST['rss_output_fulltext']) ? addslashes($_POST['rss_output_fulltext']) : 'y',
172 'isgravatar' => isset($_POST['isgravatar']) ? addslashes($_POST['isgravatar']) : 'n',
173 'comment_paging' => isset($_POST['comment_paging']) ? addslashes($_POST['comment_paging']) : 'n',
174 'comment_pnum' => isset($_POST['comment_pnum']) ? intval($_POST['comment_pnum']) : '',
175 'comment_order' => isset($_POST['comment_order']) ? addslashes($_POST['comment_order']) : 'newer',
176 'istreply' => isset($_POST['istreply']) ? addslashes($_POST['istreply']) : 'n',
177 'ischkreply' => isset($_POST['ischkreply']) ? addslashes($_POST['ischkreply']) : 'n',
178 'reply_code' => isset($_POST['reply_code']) ? addslashes($_POST['reply_code']) : 'n',
179 'index_twnum' => isset($_POST['index_twnum']) ? intval($_POST['index_twnum']) : 10,
180 'att_maxsize' => isset($_POST['att_maxsize']) ? intval($_POST['att_maxsize']) : 20480,
181 'att_type' => isset($_POST['att_type']) ? str_replace('php', 'x', strtolower(addslashes($_POST['att_type']))) : '',
182 'att_imgmaxw' => isset($_POST['att_imgmaxw']) ? intval($_POST['att_imgmaxw']) : 420,
183 'att_imgmaxh' => isset($_POST['att_imgmaxh']) ? intval($_POST['att_imgmaxh']) : 460,
184 'detect_url' => isset($_POST['detect_url']) ? addslashes($_POST['detect_url']) : 'n',
185 );
186
187 if ($getData['login_code'] == 'y' && !function_exists("imagecreate") && !function_exists('imagepng')) {
188     emMsg("开启登录验证码失败!服务器空间不支持GD图形库","configure.php");
189 }
190 if ($getData['comment_code'] == 'y' && !function_exists("imagecreate") && !function_exists('imagepng')) {
191     emMsg("开启评论验证码失败!服务器空间不支持GD图形库","configure.php");
192 }
193 if ($getData['blogurl'] && substr($getData['blogurl'], -1) != '/') {
194     $getData['blogurl'] .= '/';
195 }
196 if ($getData['blogurl'] && strncasecmp($getData['blogurl'], 'http', 4)) {
197     $getData['blogurl'] = 'http://'.$getData['blogurl'];
198 }
199
200 foreach ($getData as $key => $val) {
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221727-a3f7f1h4-1120-1.png>)

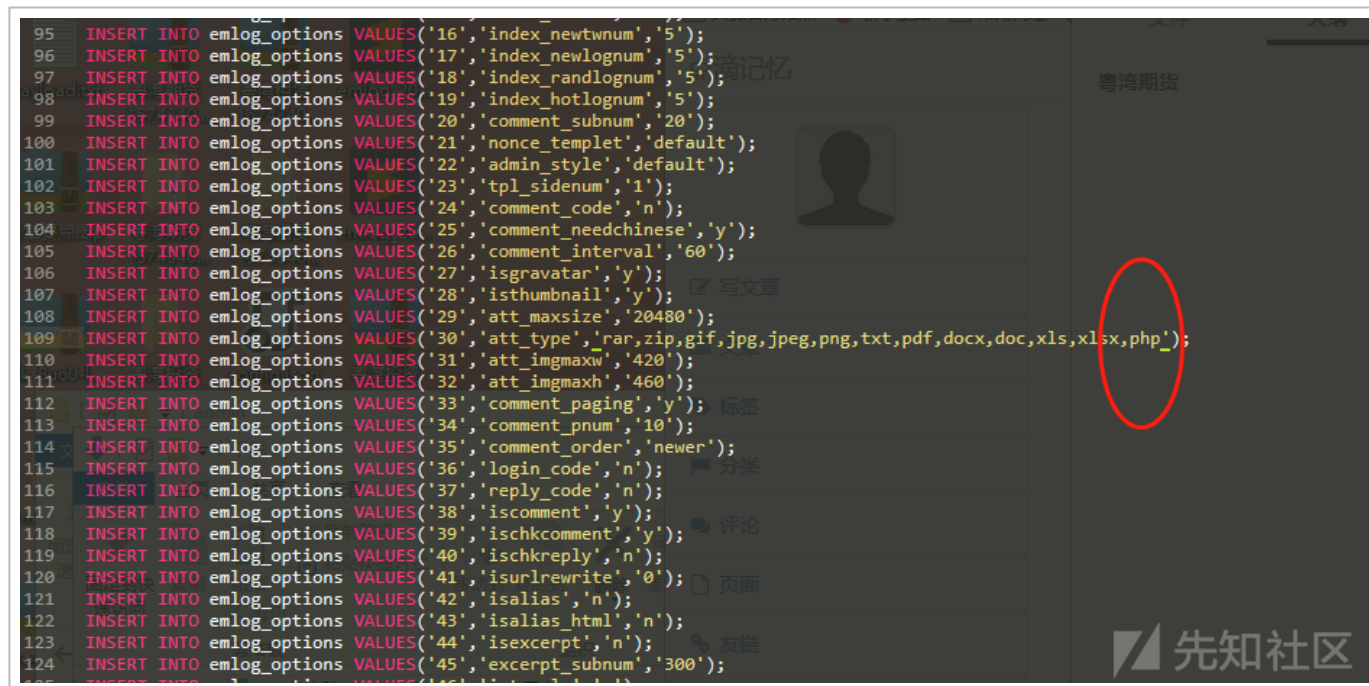
(https://xzmolany.github.io/media/apread/picture/20191127210539.jpg)

此时我注意到了备份功能，这个后缀是保存在数据库中的，既然我不能直接将后缀写入数据库，那么我不能通过数据备份恢复的方法写入数据库呢。

这里直接备份所有表，下载下来然后找到写入后缀的语句，加上 php, 如下



(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221830-c9aa83d6-1120-1.png>)



(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221813-bf9242ee-1120-1.png>)

导入备份后发现设置中的上传后缀有 php

☒ 评论分页, 每页显示  条评论, 较新的  排在前面

KB, 附件上传最大限制。

允许上传的附件类型 (多个用半角逗号)

☒ 上传图片生成缩略图, 最大尺寸:  x  (单位: 像素)


ICP备案号:

首页底部信息(支持html, 可用于添加流量统计代码):

(https://xzfile.aliyuncs.com/media/upload/picture/20191127221859-dabdf946-1120-1.png)

直接在文章发表出上传 PHP 文件即可 getshell

火狐官方网站 新手上路 常用网址 京东商城 point.mycute.cn/fun... http://www.plasf.cn/... dib2ekjwroarctf.4ho... 移动

PHP Version 5.5.38 

System	Windows NT DESKTOP-JH5305P 6.2 build 9200 (Windows 8 Home Premium Edition) i586
Build Date	Jul 20 2016 11:08:49
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	cscrip /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\x86\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\x86\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\x86\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	E:\phpStudy\PHPTutorial\php\php-5.5.38\php.ini
Scan this dir for additional .ini files	(none)

Additional .ini files parsed	(none)
PHP API	20021111

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127221927-eb3bf7fa-1120-1.png>)

## 后台 getshell (二)

我们还发现这个 cms 中上传插件的地方可以上传 zip，这里我们想如果在插件中插入一句话木马是不是也能 getshell? 我们找到插件上传的脚本，发现使用了一个叫 `emUnzip` 的函数。

```

<p>您的emlog已经安装好了，现在可以开始您的创作了，就这么简单!</p>
<p><b>用户名</b>: {$admin}</p>
<p><b>密码</b>: 您刚才设定的密码</p>";
if (DEL_INSTALLER == 1 && !@unlink('./install.php')) || (DEL_INSTALLER == 0) {
    $result .= "<p style='color:red;margin:10px 20px;'>警告：请手动删除根目录下安装文件：install.php</p> ";
}
$result .= "<p style='text-align:right;'><a href='./'>访问首页</a> | <a href='./admin/'>登录后台</a></p>";
emMsg($result, 'none');
}
?>

```

文件名	修改日期	类型	大小
config.php	2019/11/26 22:51	JetBrains PhpSto...	1
index.php	2019/11/26 21:30	JetBrains PhpSto...	1
init.php	2019/11/26 21:30	JetBrains PhpSto...	2
install.php	2019/11/26 22:53	JetBrains PhpSto...	21
robots.txt	2019/11/26 21:30	文本文档	1

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127220911-7c50b2fa-111f-1.png>)

我们继续追踪这个函数，发现这个解压函数是使用 `ZipArchive()` 类来实现解压缩的。我们可以看到代码 755 行中获取了压缩包的内部目录 / 文件的名称，并将其分割为数组将第一个元素赋值给了 `$dir`，我们看到 switch 中 `plugin` 选项，这里又将获取的第一个文件夹名称赋值给 `$plugin_name`，使用 `getFromName` 方法获取了压缩包是否存在 `$dir . $plugin_name . '.php'` 这个文件。综上这里就是检测压缩包中文件夹里面是否存在一个与文件夹名称一致的 PHP 文件，最后在再压。这里也没对文件进行其他校验操作。因此我们只要再构造 文件夹名和文件名相同的内容的压缩包，同时由上方代码也可以知道，文件将会被解压到

/content/plugins/a/a.php (文件夹名称 a)

```
0  /**
1  * 解压zip
2  * @param type $zipfile 要解压的文件 aae3d6240938911ac7257f7;
3  * @param type $path 解压到该目录
4  * @param type $type
5  * @return int
6  */
7  function emUnZip($zipfile, $path, $type = 'tpl') {
8      if (!class_exists('ZipArchive', FALSE)) {
9          return 3; //zip模块问题
10     }
11     $zip = new ZipArchive();
12     if (@$zip->open($zipfile) !== TRUE) {
13         return 2; //文件权限问题
14     }
15     $r = explode('/', $zip->getNameIndex(0), 2);
16     $dir = isset($r[0]) ? $r[0] . '/' : '';
17     switch ($type) {
18         case 'tpl':
19             $re = $zip->getFromName($dir . 'header.php');
20             if (false === $re)
21                 return -2;
22             break;
23         case 'plugin':
24             $plugin_name = substr($dir, 0, -1);
25             $re = $zip->getFromName($dir . $plugin_name . '.php');
26             if (false === $re)
27                 return -1;
28             break;
29         case 'backun':
```

越权漏洞

后台gets

后台gets

可以上传压缩包格式的插件，需要上传文件夹名和文件名相同的内容的压缩包

php, 读取



```
9         $sql_name = substr($dir, 0, -1);
10         if (getFileSuffix($sql_name) != 'sql')
11             return -3;
12         break;
13     case 'update':
14         break;
15 }
16 if (true === @$zip->extractTo($path)) {
17     $zip->close();
18     return 0;
19 } else {
20     return 1; //文件权限问题
21 }
22 }
```

ASCI Line 6, C

先知社区


(<https://xzfile.aliyuncs.com/media/upload/picture/20191127220940-8dadae7c-111f-1.png>)

构建压缩包上传插件:

www.m/content/plugins/a/a.php

新手上路 常用网址 京东商城 point.mycute.cn/fun... http://www.plasf.cn/... dib2ekjwroarctf.4ho...

### PHP Version 7.2.1



System	Windows NT DESKTOP-JH5305P 10.0 build 17134 (Windows 10) i586
Build Date	Jan 4 2018 03:59:32
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x86
Configure Command	cscrip /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=.\obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	E:\phpStudy\PHPTutorial\php\php-7.2.1-nts\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS,VC15
PHP Extension Build	API20170718,NTS,VC15

Debug Build	no
Thread Safety	disabled

(<https://xzfile.aliyuncs.com/media/upload/picture/20191127220959-992c6acc-111f-1.png>)

## 总结

这次代码审计也是十分传统地从安装文件入手，黑盒结合白盒测试的方法进行审计。总的来说作者有一点安全意识在输入时候对 php 进行过滤，但是如果服务器是 iis 可以解析 asp 但是这里并没有对 asp 进行过滤，并且如果服务器解析 phtml 这些后缀在附件上传处同样可以 getshell。总的来说在开发时候上传部分

应当锁死不应让用户可控。

## 参考

<https://blog.csdn.net/luoluozi1b/article/details/72853885>

(<https://blog.csdn.net/luoluozi1b/article/details/72853885>)