Talos Vulnerability Report

TALOS-2020-1200

Rukovoditel Project Management App application SQL injection vulnerability in the 'access_rules/rules_form' page

APRIL 8, 2021

CVE NUMBER

CVE-2020-13591

Summary

An exploitable SQL injection vulnerability exists in the ‘‘access_rules/rules_form’ page of the Rukovoditel Project Management App 2.7.2. A specially crafted HTTP request can lead to SQL injection. An attacker can make an authenticated HTTP request to trigger this vulnerability, 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 design to support project managers in complex tasks.

The `entities_id` parameter in `access_rules/rules_form` page is vulnerable to authenticated SQL injection. The following request would trigger the vulnerability:

```
POST /crm/index.php?module=access_rules/rules_form&fields_id=1&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: 173

heading_field_id=1&selected_fields=1&field_type=fieldtype_input&id=&copy_selected=1&copy_to_entities_id=1&copy_to_form_tabs_id=1&fields_in_l
isting[name]=1&name=1&fields_id=1
```

The above SQL injection exists in the `access_rules/rules_form` page due to lack of filtering applied on specific parameter. At line 78 of the source we can see that an unsanitized `entities_id` is used as part of `select` query.

78 $fields_query = db_query(“select f.*, t.name as tab_name from app_fields f, app_forms_tabs t where f.type not in (“ . fields_types::get_reserverd_types_list() . “) and f.entities_id=’” . $_GET[‘entities_id’] . “’ and f.forms_tabs_id=t.id order b y t.sort_order, t.name, f.sort_order, f.name”);

```
79 while($fields = db_fetch_array($fields_query))
80 {
81         $choices[$fields['id']] = $fields['name'];
82 }
83 ?>
```

An attacker either needs administrator privileges or they could trigger this vulnerability through cross-site request forgery.

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.

VULNERABILITY REPORTS

PREVIOUS REPORT

TALOS-2020-1201

NEXT REPORT

TALOS-2020-1199

© 2022 Cisco Systems, Inc. and/or its affiliates. All rights reserved. View our Privacy Policy.