Pre-authenticated SQL injection in GLPI <= 9.3.3

Security advisory
2019-04-29

Thomas Chauchefoin
Vulnerability description

Presentation of GLPI

“GLPI ITSM is a software for business powered by open source technologies. Take control over your IT infrastructure: assets inventory, tickets, MDM.”

The issue

Synacktiv discovered that GLPI exposes a script (/scripts/unlock_tasks.php) that not correctly sanitize user-controlled data before using it in SQL queries. Thus, an attacker could abuse the affected feature to alter the semantic original SQL query and retrieve database records. This script is reachable without authentication.

While user passwords are hashed with bcrypt (2^{10} expansion rounds) and make any bruteforce unrealistic, the Remember Me feature can be abused to reach the dashboard after leaking the field personal_token associated to a user.

Mitigation

Using prepared statements will be enough to prevent the SQL injection. In addition, it is advised to add the following snippet on top of all pages of the folder scripts/ to prevent direct access:

```php
if (php_sapi_name() !== 'cli') {
    die();
}
```

It is strongly advised to update to GLPI 9.4, since this script is not present anymore in this version.

Affected versions

The last stable version at the time of this advisory, 9.3.3, is known to be affected. It seems that commit b550fc3c787e55a7a747ff31f46157656a7c8006² first introduced the vulnerability.

The script is not present anymore in release candidates of 9.4.

Timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-02-06</td>
<td>Advisory sent to GLPI Project (<a href="mailto:glpi-security@ow2.org">glpi-security@ow2.org</a>)</td>
</tr>
<tr>
<td>2019-02-07</td>
<td>Vulnerability fixed in branch 9.3/bugfixes in 684d4fc423652ec7dde21cac4d41c2df53f56b3c³</td>
</tr>
<tr>
<td>2019-04-11</td>
<td>Fix is part of the release 9.3.4</td>
</tr>
<tr>
<td>2019-04-29</td>
<td>Publication of this advisory</td>
</tr>
</tbody>
</table>

---

1. [https://glpi-project.org/](https://glpi-project.org/)
2. [https://github.com/glpi-project/glpi/commit/b550fc3c787e55a7a747ff31f46157656a7c8006](https://github.com/glpi-project/glpi/commit/b550fc3c787e55a7a747ff31f46157656a7c8006)
3. [https://github.com/glpi-project/glpi/commit/684d4fc423652ec7dde21cac4d41c2df53f56b3c](https://github.com/glpi-project/glpi/commit/684d4fc423652ec7dde21cac4d41c2df53f56b3c)
Technical description and proof-of-concept

Accessing the script /scripts/unlock_tasks.php does not require any kind of authentication. While a .htaccess file prevents any direct access, this script is ignored by web servers other than Apache (nginx, Caddie, etc).

However, this script accepts a GET parameter named `cycle` that will be directly concatenated in a SQL query:

```php
if (isset($_GET['cycle'])) {
    $cycle = $_GET['cycle'];
} else {
    $cycle = 25;
}

$crontask = new Crontask();
$query = "SELECT `id`, `name` FROM `glpi_crontasks` WHERE `state` = '".Crontask::STATE_RUNNING."' AND unix_timestamp(`lastrun`) + $cycle * `frequency` < unix_timestamp(now())";
```

This allows change the original SQL query's behaviour and extract records from the database.

It should be noted that the exploitation is eased by the presence of an error message using results of this query; the retrieved records will be shown directly in page's body:

```php
if (isset($_GET['only_tasks'])) {
    $only_tasks = explode(',', $_GET['only_tasks']);
} else {
    $only_tasks = [];
}

foreach ($DB->request($query) as $task) {
    if (!empty($only_tasks) && !in_array($task['name'], $only_tasks)) {
        echo $task['name'] . " is still running but not in the whitelist\n"
        continue;
    }
}
```

While addslashes() is applied on $_GET / $_POST / $_REQUEST and prevent the exploitation on other endpoints, quotes are not required to perform the injection here.

As a proof, the version of the database software in use can be retrieved with the following request:

```
$ curl 'http://glpi/scripts/unlock_tasks.php?cycle=1%20UNION%20ALL%20SELECT%201,(@version)--%20&only_tasks=1'
```

Date: 2019-02-04 17:47 Start unlock script 10.1.37-MariaDB-0+deb9u1 is still running but not in the whitelist Number of unlocked tasks: 0

Impact

A successful exploitation could allow an attacker to extract records from the database and, depending on the DBMS’ permission scheme, access other databases or the local filesystem. Ultimately, by abusing the Remember Me feature, it allows accessing the dashboard with the privileges of the user glpi.