



Vulnerabilities Found in Patreon WordPress plugin

Updated on May 27, 2021 - Marc Montpas

During an internal audit of the [Patreon plugin](#) for [WordPress](#), the Jetpack Scan team found several weak points that would allow someone to take over a website.

These vulnerabilities were disclosed to the plugin authors, who promptly released version 1.7.2, which fixes all of these issues. If you're running an older version of the plugin, please update today!

Read on for all of the technical details. If this goes over your head, don't worry. We offer [Jetpack Scan](#) to handle malware scanning and automated upgrades or removal for you.

Our team identified various attack vectors, including [Local File Disclosure](#), [Cross-Site Request Forgery](#) (CSRF), and [Reflected Cross-Site Scripting](#) (XSS) vulnerabilities.

Local File Disclosure vulnerabilities are bugs that bad actors can use to gain access to critical information, such as a website's secret keys and database credentials. Reflected Cross-Site Scripting and Cross-Site Request Forgery vulnerabilities are issues that enable attackers to perform specific actions on behalf of unsuspecting users by tricking them to click carefully crafted malicious links.

If exploited, some of these could enable malicious individuals to take over vulnerable websites.

Local File Disclosure Vulnerability

Affected Versions: < 1.7.0

CVE ID: CVE-2021-24227

CVSSv3: 7.5

CWSS: 83.6

```

170 public static function servePatronOnlyImage( $image=false ) {
171
172     if ( ( !isset( $image ) OR !$image ) AND isset( $_REQUEST['patron_only_image'] ) ) {
173         $image = $_REQUEST['patron_only_image'];
174     }
175
176     if ( !$image OR $image == '' ) {
177         // This is not a rewritten image request. Exit.
178         return;
179     }
180
181     if ( !( isset( $_REQUEST['patreon_action'] ) AND $_REQUEST['patreon_action'] == 'serve_patron_only_image' ) ) {
182         return;
183     }
184
185     $upload_locations = wp_upload_dir();
186
187     // We want the base upload location so we can account for any changes to date based subfolders in case there are
188
189     $upload_dir = substr( wp_make_link_relative( $upload_locations['baseurl'] ) , 1 );
190
191     $image = get_site_url() . '/' . $upload_dir . '/' . $image;
192
193     if ( current_user_can( 'manage_options' ) ) {
194         Patreon_Protect::readAndServeImage( $image );
195     }
196
197     // Below define can be defined in any plugin to bypass core locking function and use a custom one from plugin
198     // It is independent of the plugin load order since it checks if it is defined.
199     // It can be defined by any plugin until right before the_content filter is run.
200
201     if ( apply_filters( 'ptrn/bypass_image_filtering', defined( 'PATREON_BYPASS_IMAGE_FILTERING' ) ) ) {
202         Patreon_Protect::readAndServeImage( $image );
203     }
204
205     // Check if the image is protected:
206
207     $attachment_id = attachment_url_to_postid( $image );
208
209     // attachment_url_to_postid returns 0 if it cant find the attachment post id
210

```

```

211 |         if ( $attachment_id == 0 ) {
212 |             // Couldnt determine attachment post id. Try to get id from thumbnail
213 |             $attachment_id = Patreon_Protect::getAttachmentIDfromThumbnailURL( $image );
214 |             //No go. Have to get out and serve the image normally
215 |             if ( $attachment_id == 0 OR !$attachment_id ) {
216 |                 Patreon_Protect::readAndServeImage( $image );
217 |             }
218 |         }

```

Patreon-Connect contained a local file disclosure vulnerability that could be abused by anyone visiting the site. Using this attack vector, an attacker could leak important internal files like wp-config.php, which contains database credentials and cryptographic keys used in the generation of nonces and cookies.

If successfully exploited, this security flaw could lead to a complete site takeover by bad actors.

Reflected XSS on Login Form

Affected Versions: < 1.7.2

CVE ID: CVE-2021-24228

CVSSv3: 8.8

CWSS: 80.6

```

553 | public static function processPatreonMessages() {
554 |     $patreon_error = '';
555 |     if ( isset( $_REQUEST['patreon_error'] ) ) {
556 |         // If any specific error message is sent from Patreon, prepare it
557 |         $patreon_error = ' - Patreon returned: ' . $_REQUEST['patreon_error'];
558 |     }
559 |     if ( isset( $_REQUEST['patreon_message'] ) ) {
560 |         return ' <div class="patreon_message"> ' . apply_filters( 'ptrn/error_message', self::$messages_map[ $_REQUEST['patreon_message'] ] ) . $patreon_error . ' </div>';
561 |     }
562 | }
563 |
564 |
565 |

```

Patreon-Connect hooks on the WordPress login form (wp-login.php) and offers to allow users to authenticate on the site using their Patreon account.

Unfortunately, some of the error logging logic behind the scene allowed user-controlled input to be reflected on the login page, unsanitized.

To successfully exploit this vulnerability, an attacker needs to trick his victim into visiting a booby-trapped link containing malicious Javascript code. Since Javascript runs in the victim's browser context, an attacker can adjust the code hidden in that link to do whatever this user's privileges allow him to.

If this attack succeeds against an administrator, the script can completely take over the site.

Reflected XSS on AJAX action 'patreon_save_attachment_patreon_level'

Affected Versions: < 1.7.2

CVE ID: CVE-2021-24229

CVSSv3: 8.8

CWSS: 80.6

```

698 | $args = array (
699 |     'attachment_id' => $attachment_id,
700 |     'patreon_level' => $_REQUEST['patreon_attachment_patreon_level'],
701 |     'message' => $message,
702 | );
703 |
704 | echo self::make_image_lock_interface( $args );
705 |
706 |
707 | public function make_image_lock_interface( $args = array() ) {
708 |     $interface = '';
709 |     $interface .= '<div class="patreon_image_lock_modal_content">';
710 |     $interface .= '<span class="patreon_image_lock_modal_close">&times;</span>';
711 |     $interface .= '<form id="patreon_attachment_patreon_level_form" action="/wp-admin/admin-ajax.php" method="post">';
712 |     $interface .= '<h1 class="patreon_image_locking_interface_heading">Lock Image</h1>';
713 |     $interface .= '<div class="patreon_image_locking_interface_level">';
714 |     $interface .= '<span class="patreon_image_locking_interface_input_prefix"><input id="patreon_attachment_patreon_level" type="text" name="patr

```

The plugin also uses an AJAX hook to update the pledge level required by Patreon subscribers to access a given attachment. This action is accessible for user accounts with the 'manage_options' privilege (i.e., only administrators).

Unfortunately, one of the parameters used in this AJAX endpoint is not sanitized before being printed back to the user, so the risk it represents is the same as the previous XSS vulnerability we described.

CSRF Allowing Attackers To Overwrite/Create User Meta

Affected Versions: < 1.7.0

CVE ID: CVE-2021-24230

CVSSv3: 6.5

CWSS: 42

```

1331 | public function toggle_option() {
1332 |     if ( !is_admin() && current_user_can( 'manage_options' ) ) {
1333 |         return;
1334 |     }
1335 |     $current_user = wp_get_current_user();
1336 | }
1337 |
1338 |

```

```

1339     $option_to_toggle = $_REQUEST['toggle_id'];
1340
1341     $current_value = get_user_meta( $current_user->ID, $option_to_toggle, true );
1342
1343     $new_value = 'off';
1344
1345     if( !$current_value OR $current_value == 'off' ) {
1346         $new_value = 'on';
1347     }
1348
1349     update_user_meta( $current_user->ID, $option_to_toggle, $new_value );
1350
1351 }

```

Some endpoints did not validate the request it received was sent following a legitimate action from a user, [which you can do using nonces](#). One of these unprotected endpoints allowed malevolent individuals to craft a booby-trapped link that would overwrite or create arbitrary user metadata on the victim's account once visited.

If exploited, this bug can be used to overwrite the “wp_capabilities” meta, which contains the affected user account's roles and privileges. Doing this would essentially lock them out of the site, blocking them from accessing paid content.

CSRF Allowing Attackers To Disconnect Sites From Patreon

Affected Versions: < 1.7.0

CVE ID: CVE-2021-24231

CVSSv3: 6.5

CWSS: 26.1

```

2338 if ( isset( $_REQUEST['patreon_wordpress_action'] ) AND $_REQUEST['patreon_wordpress_action'] == 'disconnect_site_from_patreon' AND is_admin() AND
2339
2340 // Admin side, user is admin level. Perform action:
2341
2342 // To disconnect the site from a particular creator account, we will delete all options related to creator account, but we will leave other plugin
2343
2344 $options_to_delete = array(
2345     'patreon-custom-page-name',
2346     'patreon-fetch-creator-id',
2347     'patreon-creator-tiers',
2348     'patreon-creator-last-name',
2349     'patreon-creator-first-name',
2350     'patreon-creator-full-name',
2351     'patreon-creator-url',
2352     'patreon-campaign-id',
2353     'patreon-creators-refresh-token-expiration',
2354     'patreon-creator-id',
2355     'patreon-setup-wizard-last-call-result',
2356     'patreon-creators-refresh-token',
2357     'patreon-creators-access-token',
2358     'patreon-client-secret',
2359     'patreon-client-id',
2360     'patreon-setup_is_being_done',
2361     'patreon-setup-done',
2362     'patreon-currency-sign',
2363 );
2364
2365 // Ask the API to delete this client:
2366
2367 $creator_access_token = get_option( 'patreon-creators-access-token', false );
2368 $client_id            = get_option( 'patreon-client-id', false );
2369
2370 // Exceptions until v1 v2 transition is complete
2371
2372 $api_version = get_option( 'patreon-installation-api-version' );
2373
2374 if ( $api_version == '1' ) {
2375
2376     // Delete override - proceed with deleting local options
2377
2378     foreach ( $options_to_delete as $key => $value ) {
2379         delete_option( $options_to_delete[$key] );
2380     }
2381
2382     update_option( 'patreon-installation-api-version', '2' );
2383     update_option( 'patreon-can-use-api-v2', true );
2384
2385     wp_redirect( admin_url( 'admin.php?page=patreon_wordpress_setup_wizard&setup_stage=reconnect_0' ) );
2386     exit;
2387 }
2388

```

This one is similar to the last vulnerability in that it's the same kind of attack (CSRF) but is targeting administrators. This particular attack vector works like the previous. An attacker needs to get a logged-in administrator to visit a specially crafted link.

Since this specific endpoint can disconnect a site from Patreon, attackers targeting this attack vector can also do just that, which would prevent new content from being synchronized to the site.

Timeline

- Initial contact attempt (unsuccessful) – Dec 4th
- Second contact attempt – Dec. 11th
- Authors acknowledge the report – Dec. 15th
- Version 1.7.0 is released – Jan 5th
- We report two additional XSS issues – March 9th
- Authors acknowledge the second report – March 9th
- Version 1.7.2 is released – March 11th

Conclusion

We recommend that you check the current version of the Patreon-Connect plugin you are using on your site and, if not 1.7.2, update it as soon as possible!

At Jetpack, we work hard to make sure your [websites are protected from these types of vulnerabilities](#). To stay one step ahead of any new threats, check out [Jetpack Scan](#), which includes security scanning and automated malware removal.

Credits

This security disclosure was made possible thanks to George Stephanis, Fioravante Souza, Miguel Neto, Benedict Singer and Marc Montpas.

This entry was posted in [Security](#), [Vulnerabilities](#). Bookmark the [permalink](#).



Marc Montpas

Marc's interests led him to work in the trenches of cybersecurity for the better part of the last decade, notably at companies like Sucuri and GoDaddy. His journey led him to uncover several high-impact security issues while auditing open-source platforms, like WordPress. He's an avid Hacker Capture The Flag player and loves to hypothesize new attack vectors.

Explore the benefits of Jetpack

Learn how Jetpack can help you protect, speed up, and grow your WordPress site.

[Compare plans](#)

Have a question?

Comments are closed for this article, but we're still here to help! Visit the support forum and we'll be happy to answer any questions.

[View support forum](#)

Search

Get news & tips from Jetpack

Enter your email address to follow this blog and receive news and updates from Jetpack!

Email Address

[Subscribe](#)

Join 111,148 other subscribers

Browse by Topic

[Affiliates](#) (1)

[Analytics](#) (6)

[Code snippets](#) (32)

[Contribute](#) (6)

[Customer Stories](#) (6)

[Ecommerce](#) (11)

[Events](#) (5)

[Features](#) (56)

[Grow](#) (11)

[hosting](#) (1)

[Innovate](#) (6)

[Jetpack News](#) (45)

[Learn](#) (65)

[Meet Jetpack](#) (14)

- [Performance \(24\)](#)
- [Photos & Videos \(9\)](#)
- [Promotions \(2\)](#)
- [Releases \(166\)](#)
- [Search Engine Optimization \(12\)](#)
- [Security \(75\)](#)
- [Small Business \(16\)](#)
- [Social Media \(13\)](#)
- [Support Stories \(3\)](#)
- [Tips & Tricks \(85\)](#)
- [Uncategorized \(5\)](#)
- [Utilities & Maintenance \(4\)](#)
- [Vulnerabilities \(18\)](#)
- [Website Design \(13\)](#)
- [WordAds \(1\)](#)
- [WordCamp \(3\)](#)



EN

WordPress Plugins

- [Akismet Anti-spam](#)
- [Jetpack](#)
- [Jetpack Boost](#)
- [Jetpack CRM](#)
- [Jetpack Protect](#)
- [Jetpack Search](#)
- [Jetpack Social](#)
- [Jetpack VideoPress](#)
- [VaultPress Backup](#)
- [WP Super Cache](#)

Partners

- [Recommended Hosts](#)
- [For Hosts](#)
- [For Agencies](#)

Developers

- [Documentation](#)
- [Beta Program](#)
- [Contribute to Jetpack](#)

Legal

- [Terms of Service](#)
- [Privacy Policy](#)
- [GDPR](#)
- [Privacy Notice for California Users](#)

Help

- [Knowledge Base](#)
- [Forums](#)
- [Security Library](#)
- [Contact Us](#)
- [Press](#)

Social

Mobile Apps

