Multiple vulnerabilities in WP Fastest Cache plugin

*Posted on **October 14, 2021** by Marc Montpas*
During an internal audit of the WP Fastest Cache plugin, we uncovered an Authenticated SQL Injection vulnerability and a Stored XSS (Cross-Site Scripting) via Cross-Site Request Forgery (CSRF) issue.

If exploited, the SQL Injection bug could grant attackers access to privileged information from the affected site's database (e.g., usernames and hashed passwords). It can only be exploited if the classic-editor plugin is also installed and activated on the site.

Successfully exploiting the CSRF & Stored XSS vulnerability could enable bad actors to perform any action the logged-in administrator they targeted is allowed to do on the targeted site.

We reported the vulnerabilities to this plugin's author via email, and they recently released version 0.9.5 to address them. We strongly recommend that you update to the latest version of the plugin and have an established security solution on your site, such as Jetpack Security.

Details

Plugin Name: WP Fastest Cache
Plugin URI: https://wordpress.org/plugins/wp-fastest-cache/
Author: https://www.wpfastestcache.com/

The Vulnerabilities

Authenticated SQL Injection

Affected versions: < 0.9.5
CVE-ID: CVE-2021-24869
CVSSv3.1: 7.7
CWSS: 73.6

```
    public static
    function
    set_urls_with_terms
161 (){
162         global $wpdb;
163         $terms = $wpdb->get_results("SELECT * FROM `".$wpdb-
164 >prefix."term_relationships` WHERE `object_id`=".static::$id, ARRAY_A);
165
166         foreach ($terms as $term_key => $term_val){
167                 static::set_term_urls($term_val["term_taxonomy_id"]);
168         }
}
```

The `set_urls_with_terms` method directly concatenates **static::$id** to an SQL query, which is an issue since any logged-in users can store arbitrary values in that property, via the SinglePreloadWPFC::set_id() method. This method is executed when the `admin_notices` WordPress action is run.

```
115
116 public static function set_id(){
117         if(isset($_GET["post"]) && $_GET["post"]){
118                 static::$id = esc_sql($_GET["post"]);
119
120                 if(get_post_status(static::$id) != "publish"){
121                         static::$id = 0;
122                 }
123         }
}
```

Although `set_id` checks that the provided ID points to a valid, published post using the get_post_status() function, this isn't enough to validate that it *only* contains that ID.

The get_post_status() function eventually uses get_post() internally, which casts the ID it receives to integer before querying the database for the related post.

Simply put, if the ID provided is `1234 OR 1=1`, get_post_status**()** will retrieve the status of the post whose ID is `1234`, but `1234 OR 1=1` is going to be concatenated to the vulnerable SQL query in SinglePreloadWPFC::set_urls_with_terms().

Stored XSS Via CSRF

Affected versions: < 0.9.5
CVE-ID: CVE-2021-24869
CVSSv3.1: 9.6
CWSS: 74.7

The CdnWPFC::save_cdn_integration() method is used by the `wp_ajax_wpfc_save_cdn_integration` AJAX action to set-up CDN-specific options. While it did perform privilege checks like current_user_can() to ensure whoever sent that request is allowed to change those settings, it did not validate that they *intended* to, which is what nonce checks do.

Furthermore, we discovered that attackers could potentially abuse some of these options to store rogue Javascript on the affected website.

Timeline

The authors were initially reluctant to acknowledge the CSRF issue, but after obtaining a second opinion from the WordPress plugin team, they fixed it in version 0.9.5.

2021-09-28 – Initial contact with WP Fastest Cache
2021-09-29 – We send them details about these vulnerabilities
2021-10-01 – We share with them a video proof of concept to demonstrate the risk CSRF pose
2021-10-01 – We reach out to the WordPress plugin team for help
2021-10-11 – WP Fastest Cache 0.9.5 is released

Conclusion

We recommend that you check which version of the WP Fastest Cache plugin your site is using, and if it is less than 0.9.5, update it as soon as possible!

At Jetpack, we work hard to make sure your websites are protected from these types of vulnerabilities. We recommend that you have a security plan for your site that includes malicious file scanning and backups. Jetpack Security is one great WordPress security option to ensure your site and visitors are safe.

Credits

Original researcher: Marc Montpas

Thanks to the rest of the Jetpack Scan team for feedback, help, and corrections.

*This entry was posted in **scan**, **Security**, **Vulnerabilities** and tagged **Jetpack**, **scan**, **Security**. Bookmark the **permalink**.*