
# Severe Vulnerabilities Fixed in All In One SEO Plugin Version 4.1.5.3

*Updated on December 14, 2021 - Marc Montpas*

During an internal audit of the All In One SEO plugin, we uncovered an SQL Injection vulnerability and a Privilege Escalation bug.

If exploited, the SQL Injection vulnerability could grant attackers access to privileged information from the affected site's database (e.g., usernames and hashed passwords).

The Privilege Escalation bug we discovered may grant bad actors access to protected REST API endpoints they shouldn't have access to. This could ultimately enable users with low-privileged accounts, like subscribers, to perform remote code execution on affected sites.

We reported the vulnerabilities to the plugin's author via email, and they recently released version 4.1.5.3 to address them. We strongly recommend that you update to the latest plugin version and have an established security solution on your site, such as Jetpack Security.

## Details

Plugin Name: All In One SEO
Plugin URI: https://wordpress.org/plugins/all-in-one-seo-pack/
Author: https://aioseo.com/

## The Vulnerabilities

### Authenticated Privilege Escalation

**Affected versions:** Every version between 4.0.0 and 4.1.5.2 inclusively.
**CVE-ID:** CVE-2021-25036
**CVSSv3.1:** 9.9
**CWSS:** 92.1

```php
/**
 * Validates access from the routes array.
 *
 * @since 4.0.0
 *
 * @param  \WP_REST_Request $request The REST Request.
 * @return bool                      True if validated, false if not.
 */
public function validateAccess( $request ) {
    $route     = str_replace( '/' . $this->namespace . '/', '', $request->get_route() );
    $routeData = isset( $this->getRoutes()[ $request->get_method() ][ $route ] ) ? $this->getRoutes()[ $request->get_method() ][ $route ] : [];

    // No direct route name, let's try the regexes.
    if ( empty( $routeData ) ) {
        foreach ( $this->getRoutes()[ $request->get_method() ] as $routeRegex => $routeInfo ) {
            $routeRegex = str_replace( '@', '\@', $routeRegex );
            if ( preg_match( "@{$routeRegex}@", $route ) ) {
                $routeData = $routeInfo;
                break;
            }
        }
    }

    if ( empty( $routeData['access'] ) ) {
        return true;
    }

    // We validate with any of the access options.
    if ( ! is_array( $routeData['access'] ) ) {
        $routeData['access'] = [ $routeData['access'] ];
    }
    foreach ( $routeData['access'] as $access ) {
        if ( current_user_can( $access ) ) {
            return true;
        }
    }
```

```
255
256     if ( current_user_can( apply_filters( 'aioseo_manage_seo', 'aioseo_manage_seo' ) ) ) {
257         return true;
258     }
259
260     return false;
261 }
```

The privilege checks applied by All In One SEO to secure REST API endpoints contained a very subtle bug that could've granted users with low-privileged accounts (like subscribers) access to every single endpoint the plugin registers.

The Api::validateAccess() method relies on the REST API route being requested to know which privilege checks to enforce on a given request. Since it didn't account for the fact that WordPress treats REST API routes as case-insensitive strings, changing a single character to uppercase would completely bypass the privilege checks routine.

This is particularly worrying because some of the plugin's endpoints are pretty sensitive. For example, the `aioseo/v1/htaccess` endpoint can rewrite a site's .htaccess with arbitrary content. An attacker could abuse this feature to hide .htaccess backdoors and execute malicious code on the server.

### Authenticated SQL Injection

**Affected versions:** Every version between 4.1.3.1 and 4.1.5.2 inclusively.
**CVE-ID:** CVE-2021-25037
**CVSSv3.1:** 7.7
**CWSS:** 80.4

```
17  /**
18   * Searches for posts or terms by ID/name.
19   *
20   * @since 4.0.0
21   *
22   * @param  \WP_REST_Request  $request The REST Request
23   * @return \WP_REST_Response          The response.
24   */
25  public static function searchForObjects( $request ) {
26      $body = $request->get_json_params();
27
28      if ( empty( $body['query'] ) ) {
29          return new \WP_REST_Response( [
30              'success' => false,
31              'message' => 'No search term was provided.'
32          ], 400 );
33      }
34      if ( empty( $body['type'] ) ) {
35          return new \WP_REST_Response( [
36              'success' => false,
37              'message' => 'No type was provided.'
38          ], 400 );
39      }
40
41      $searchQuery = aioseo()->db->db->esc_like( $body['query'] );
42
43      $objects        = [];
44      $dynamicOptions = aioseo()->dynamicOptions->noConflict();
45      if ( 'posts' === $body['type'] ) {
46
47          $postTypes = aioseo()->helpers->getPublicPostTypes( true );
48          foreach ( $postTypes as $postType ) {
49              // Check if post type isn't noindexed.
50              if ( $dynamicOptions->searchAppearance->postTypes->has( $postType ) && ! $dynamicOptions->searchAppearance->postTypes->$postType->show )
51                  $postTypes = aioseo()->helpers->unsetValue( $postTypes, $postType );
52              }
53          }
54
55          $objects = aioseo()->db
56              ->start( 'posts' )
57              ->select( 'ID, post_type, post_title, post_name' )
58              ->whereRaw( "( post_title LIKE '%{$searchQuery}%' OR post_name LIKE '%{$searchQuery}%' OR ID = '{$searchQuery}' )" )
59              ->whereIn( 'post_type', $postTypes )
60              ->whereIn( 'post_status', [ 'publish', 'draft', 'future', 'pending' ] )
61              ->orderBy( 'post_title' )
62              ->limit( 10 )
63              ->run()
64              ->result();
```

The PostsTerms::searchForObjects() method, which is accessible via the `/wp-json/aioseo/v1/objects` REST API route *only* escaped user input using wpdb::esc_like() before appending said input to an SQL query. Since this method *isn't designed* to escape quotes, an attacker could still inject them and force the query to leak sensitive information from the database, like user credentials.

While this endpoint wasn't meant to be accessible to users with low-privileged accounts, the aforementioned privilege escalation attack vector made it possible for them to abuse this vulnerability.

## Timeline

2021-12-01 – Initial contact with All In One SEO
2021-12-02 – We send them details about these vulnerabilities
2021-12-08 – All In One SEO 4.1.5.3 is released

## Conclusion

We recommend that you check which version of the All In One SEO plugin your site is using, and if it is within the affected range, 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 **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.

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)

Jetpack

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**