# Unauthenticated Information Disclosure (IDOR) via Multisite switch_to_blog in My Calendar

High **joedolson** published **GHSA-2mvx-f5qm-v2ch** 4 hours ago

**Package**

**my-calendar** (Wordpress)

**Affected versions**

< 3.7.7

**Patched versions**

None

**Description**

## Summary

An unauthenticated Insecure Direct Object Reference (IDOR) and Denial of Service (DoS) vulnerability in the My Calendar plugin allows any unauthenticated user to extract calendar events (including private or hidden ones) from any sub-site on a WordPress Multisite network. On standard Single Site WordPress installations, this same endpoint crashes the PHP worker thread, creating an unauthenticated Denial of Service (DoS) vector.

## Details

The vulnerability stems from the `mc_ajax_mcjs_action AJAX` function, which handles the `mcjs_action` endpoint. This endpoint is explicitly registered for unauthenticated users:

```
<?php
// In my-calendar-ajax.php
add_action( 'wp_ajax_nopriv_mcjs_action', 'mc_ajax_mcjs_action' );
```

When the behavior parameter is set to loadupcoming, the plugin accepts an args parameter from the $_REQUEST array. Instead of validating specific expected arguments, the plugin unsafely passes the entire string into PHP's parse_str() function:

```php
<?php
$request = isset( $_REQUEST['args'] ) ? wp_unslash( sanitize_text_field( $_REQUEST['args'] ) ) : '';
$request = str_replace( '|', '&', $request );
$request = parse_str( $request, $args );
// ...
$response = my_calendar_upcoming_events( $args );
```

This allows an attacker to inject arbitrary key-value pairs into the $args array. This array is then passed to the my_calendar_upcoming_events() function located in my-calendar-widgets.php.

At the beginning of this function, the plugin processes the attacker-controlled site argument:

```php
<?php
// In my-calendar-widgets.php
if ( $args['site'] ) {
    $args['site'] = ( 'global' === $args['site'] ) ? BLOG_ID_CURRENT_SITE : $args['site'];
    switch_to_blog( $args['site'] );
}
```

The plugin blindly passes the attacker's supplied site ID into WordPress core's `switch_to_blog()` function without checking if the requesting user has the appropriate network-level privileges (e.g., Super Admin).

On Multisite configurations, the database context switches to the targeted sub-site, queries its events, and returns the HTML-rendered events array in the JSON response, leading to Information Disclosure across tenant boundaries.
On Single Site configurations, the switch_to_blog() function does not exist in WordPress core. Calling it triggers an Uncaught PHP Error (Call to undefined function switch_to_blog()), resulting in a 500 Internal Server error ("Critical Error"). Repeated requests to this unauthenticated endpoint easily exhaust server resources.

### PoC

## 1. Multisite Information Disclosure - IDOR

```
curl -s "http://<target-domain>/wp-admin/admin-ajax.php?
action=mcjs_action&behavior=loadupcoming&args=site=2"
```

## 2. Single Site Denial of Service (DoS)

If the WordPress instance is not a Multisite, passing any truthy value to the site parameter will instantly crash the request thread:

```
curl -i -s "http://<target-domain>/wp-admin/admin-ajax.php?
action=mcjs_action&behavior=loadupcoming&args=site=1"
```

## Impact

**Vulnerability Type**: Insecure Direct Object Reference (IDOR) / Information Exposure / Denial of Service (DoS)
**Who is impacted**: All sites running the "My Calendar" plugin.

Anonymous internet users can silently map the network and extract private, unpublished, or intranet-specific events from unlaunched/internal sub-sites.
Standard Single Site users are vulnerable to an easy-to-execute application-layer DoS, as it costs an attacker negligible resources to constantly crash PHP worker threads at an unauthenticated endpoint.

### Severity

High

### CVE ID

CVE-2026-40308

### Weaknesses

No CWEs

### Credits

**minhi1** Reporter