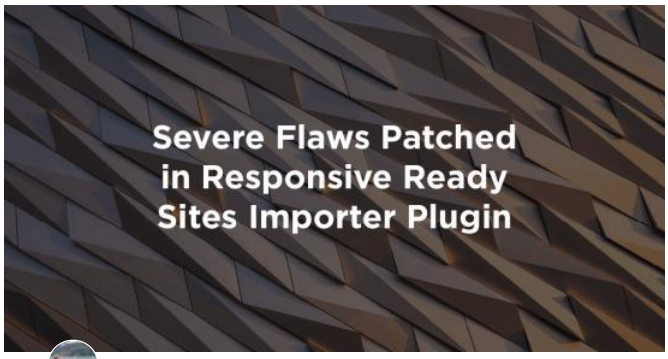

Chloe Chamberland

March 18, 2020

Severe Flaws Patched in Responsive Ready Sites Importer Plugin

On March 2nd, our Threat Intelligence team discovered several vulnerable endpoints in Responsive Ready Sites Importer, a WordPress plugin installed on over 40,000 sites. These flaws allowed any authenticated user, regardless of privilege level, the ability to execute various AJAX actions that could reset site data, inject malicious JavaScript in pages, modify theme customizer data, import .xml and .json files, and activate plugins, among many other actions.

We reached out to the plugin's developer on March 3, 2020, and they were proactive and quick to respond. They released patches consisting of nonce and permissions checks on nearly all of the AJAX endpoints before we sent over the full vulnerability details the following morning. We still provided the full disclosure, and pointed out a few AJAX endpoints missed in their initial release. They released a final patch just a few days later.

This is considered a severe security issue that could lead to attackers completely taking over WordPress sites. We highly recommend updating to the latest version available, 2.2.7, immediately.

Wordfence Premium customers received a new firewall rule on March 2, 2020, to protect against exploits targeting this vulnerability. Free Wordfence users will receive the rule after thirty days, on April 1, 2020.

Description: Unprotected AJAX Actions
Affected Plugin: Responsive Ready Sites Importer
Plugin Slug: responsive-add-ons
Affected Versions: <= 2.2.5
CVE ID: CVE-2020-12073
CVSS Score: 9.1 (Critical)
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:H/A:L
Fully Patched Version: 2.2.6

Gutenberg & Elementor Templates Importer For Responsive, also called Responsive Ready Sites Importer, is a plugin designed to import templates and site content to be used with the Gutenberg or Elementor page builders. The plugin is very simple to use and provides a plethora of templates for site owners to choose from.

The import functionality relies on various AJAX actions, with functionalities ranging from resetting site data prior to an import all the way to importing .xml and .json files to provide data for the import. We discovered 23 vulnerable endpoints, and the majority of these were found in the /class-responsive-ready-sites-importer.php file.

```
/**
 * Constructor.
 *
 * @since 1.0.0
 */
public function __construct() {

    add_action( 'init', array( $this, 'load_importer' ) );

    $responsive_ready_sites_importers_dir = plugin_dir_path( __FILE__ );
    require_once $responsive_ready_sites_importers_dir . 'class-responsive-ready-sites-importer-log.php';
    include_once $responsive_ready_sites_importers_dir . 'class-responsive-ready-sites-widgets-importer.php';
    include_once $responsive_ready_sites_importers_dir . 'class-responsive-ready-sites-options-importer.php';

    // Import AJAX.
    add_action( 'wp_ajax_responsive-ready-sites-import-set-site-data-free', array( $this, 'import_start' ) );
    add_action( 'wp_ajax_responsive-ready-sites-import-xml', array( $this, 'import_xml_data' ) );
    add_action( 'wp_ajax_responsive-ready-sites-import-wpforms', array( $this, 'import_wpforms' ) );
    add_action( 'wp_ajax_responsive-ready-sites-import-customizer-settings', array( $this, 'import_customizer_settings'
    add_action( 'wp_ajax_responsive-ready-sites-import-widgets', array( $this, 'import_widgets' ) );
    add_action( 'wp_ajax_responsive-ready-sites-import-options', array( $this, 'import_options' ) );
    add_action( 'wp_ajax_responsive-ready-sites-import-end', array( $this, 'import_end' ) );

    add_action( 'responsive_ready_sites_import_complete', array( $this, 'clear_cache' ) );

    include_once $responsive_ready_sites_importers_dir . 'batch-processing/class-responsive-ready-sites-batch-processin

    // Reset Customizer Data.
    add_action( 'wp_ajax_responsive-ready-sites-reset-customizer-data', array( $this, 'reset_customizer_data' ) );
    add_action( 'wp_ajax_responsive-ready-sites-reset-site-options', array( $this, 'reset_site_options' ) );
    add_action( 'wp_ajax_responsive-ready-sites-reset-widgets-data', array( $this, 'reset_widgets_data' ) );

    // Reset Post & Terms.
    add_action( 'wp_ajax_responsive-ready-sites-delete-posts', array( $this, 'delete_imported_posts' ) );
    add_action( 'wp_ajax_responsive-ready-sites-delete-wp-forms', array( $this, 'delete_imported_wp_forms' ) );
    add_action( 'wp_ajax_responsive-ready-sites-delete-terms', array( $this, 'delete_imported_terms' ) );

    if ( version_compare( get_bloginfo( 'version' ), '5.0.0', '>=' ) ) {
        add_filter( 'http_request_timeout', array( $this, 'set_timeout_for_images' ), 10, 2 );
    }
}
```

Using the `import_start` function tied to the `wp_ajax_responsive-ready-sites-import-set-site-data-free` action as an example below. It can be shown that there was a lack of capability checks and nonce checks as part of the functions. This was evident in all of the identified functions triggered by the registered AJAX actions that we found vulnerable.

```
/**
 * Start Site Import
```

```
public function import_start() {

        $demo_api_uri = isset( $_POST['api_url'] ) ? esc_url( $_POST['api_url'] ) : ''; //phpcs:ignore

    if ( ! empty( $demo_api_uri ) ) {

        $demo_data = self::get_responsive_single_demo( $demo_api_uri );
        if ( ! $demo_data['success'] ) {
            wp_send_json( $demo_data );
        }

        update_option( 'responsive_ready_sites_import_data', $demo_data );

        if ( is_wp_error( $demo_data ) ) {
            wp_send_json_error( $demo_data->get_error_message() );
        } else {
            do_action( 'responsive_ready_sites_import_start', $demo_data, $demo_api_uri );
        }

        wp_send_json_success( $demo_data );

    } else {
        wp_send_json_error( __( 'Request site API URL is empty. Try again!', 'responsive-addons' ) );
    }

}
```

All of the vulnerable actions could be called with a simple request to `/wp-admin/admin-ajax.php?action=[Vulnerable-Action]` along with the appropriate parameters set, by any authenticated user, including users with minimal subscriber-level permissions.

Fortunately, in the latest version of this plugin, capability checks to help control access and execution, as well as CSRF protection using WordPress nonces, were implemented on all of these endpoints.

A Deeper Dive on a Few Endpoints

Although there were several unprotected endpoints, a few were a little more worrisome than others.

The AJAX action `wp_ajax_responsive-ready-sites-import-xml` triggers a function that imports an XML file to be used to supply data as part of the import process. Then, the AJAX action `wp_ajax_responsive-wxr-import` would trigger the function that imports all the data from the previously imported XML file. Using these two actions together could allow an attacker to import an XML file containing malicious payloads such as new pages on the site. The malicious payloads would then be executed anytime a user browsed to the newly imported page. This could result in malicious site redirects and rogue administrative user creation, among other consequences.

The AJAX actions `wp_ajax_responsive-ready-sites-import-options`, `wp_ajax_responsive-ready-sites-import-widgets`, and `wp_ajax_responsive-ready-sites-import-customizer-settings` triggered functions that would import widgets, site options, and site customizer data. These could be used by an attacker to overwrite site data with malicious data of their choice.

A Brief Note To Site Owners and WordPress Developers

Site owners. Vulnerable AJAX endpoints are, unfortunately, a very common vulnerability among WordPress plugins and themes. We highly recommend disabling user registration on your site if it is not necessary for the site's functionality. If your site is running a plugin or theme with a vulnerable AJAX endpoint, this will prohibit any attackers from being able to register an account, login, and then execute attacks against these vulnerable endpoints that could potentially compromise your site.

It is also highly recommended to ensure your plugins and themes are up to date at all times as these vulnerabilities are often immediately discovered and patched. In the cases where a patch isn't released quickly, it is important you have a Web Application Firewall in place, such as the one provided by Wordfence, to help provide protection during the interim period where a vulnerability might be discovered and actively attacked before it has been completely patched.

Developers. It is incredibly important to add capability checks and CSRF protection on functions controlled by AJAX actions in plugins and themes. Subscriber-level users and above have the ability to execute these actions if the proper security measures are not in place. Many WordPress sites allow open registration, creating a large attack surface for these vulnerabilities that are typically very easy to exploit.

Use functions like `current_user_can()` to check for user capability on actions along with `wp_create_nonce()` and `wp_verify_nonce()` to verify the legitimacy of a request's source to protect against CSRF on all AJAX functions.

To see how common these vulnerabilities are, you can review some of our recently discovered unprotected AJAX actions vulnerabilities in Popup Builder, Import Export WordPress Users, 301 Redirects – Easy Redirect Manager, and RegistrationMagic. As a plugin developer, it is important to take preventive steps against creating these vulnerabilities, just as it is important to protect yourself against these as a site owner.

Disclosure Timeline

March 2, 2020 – Initial discovery and analysis of vulnerability. We release a firewall rule for Wordfence Premium customers.
March 3, 2020 – We make our initial contact attempt with the plugin development team. Developer responds and confirms that we have reached them through the appropriate inbox.
March 4, 2020 – We send over the full disclosure details. Developer responds and indicates all vulnerabilities have been fixed.
March 4-5, 2020 – We further analyze the fixes and discover a few AJAX actions left unprotected. We notify the developer.
March 11, 2020 – Developer releases final sufficient patch.
April 1, 2020 – Free Wordfence users receive firewall rule.

Conclusion

In today's post, we detailed several flaws related to unprotected AJAX actions in the Responsive Ready Sites Importer plugin. These flaws have been fully patched in version 2.2.6. We recommend that users update to the latest version available immediately. Sites running Wordfence Premium have been protected from attacks against this vulnerability since March 2, 2020. Sites running the free version of Wordfence will receive the firewall rule update on April 1, 2020.
Did you enjoy this post? Share it!

Comments

No Comments

happens.

By checking this box I agree to the terms of service and privacy policy.*

SIGN UP

Our business hours are 9am-8pm ET, 6am-5pm PT and 2pm-1am UTC/GMT excluding weekends and holidays.
Response customers receive 24-hour support, 365 days a year, with a 1-hour response time.

Stay Updated

Sign up for news and updates from our panel of experienced security professionals.

you@example.com

By checking this box I agree to the terms of service and privacy policy.*

SIGN UP