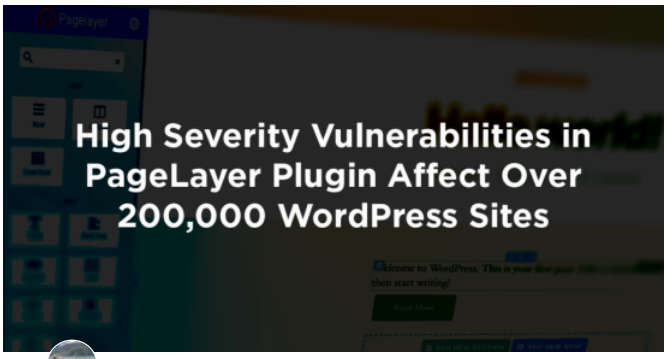


Chloe Chamberland

May 28, 2020

High Severity Vulnerabilities in PageLayer Plugin Affect Over 200,000 WordPress Sites

A few weeks ago, our Threat Intelligence team discovered several vulnerabilities present in Page Builder: PageLayer – Drag and Drop website builder, a WordPress plugin actively installed on over 200,000 sites. The plugin is from the same creators as wpCentral, a plugin within which we recently discovered a privilege escalation vulnerability.

One flaw allowed any authenticated user with subscriber-level and above permissions the ability to update and modify posts with malicious content, amongst many other things. A second flaw allowed attackers to forge a request on behalf of a site's administrator to modify the settings of the plugin which could allow for malicious Javascript injection.

We initially reached out to the plugin's developer on April 30, 2020 and after establishing an appropriate communication channel, we provided the full disclosure on May 1, 2020. They responded quickly on May 2, 2020 letting us know that they were beginning to work on fixes. An initial patch was released on May 2, 2020 and an optimal patch was released on May 6, 2020.

These are considered high-level security issues that could potentially lead to attackers wiping your site's content or taking over your site. We highly recommend an immediate update to the latest version available at the time of this publication, **which is version 1.1.4**.

Wordfence Premium customers received a new firewall rule on April 30, 2020, to protect against exploits targeting this vulnerability. Free Wordfence users will receive this rule after thirty days, on May 30, 2020.

Description: Unprotected AJAX and Nonce Disclosure to Stored Cross-Site Scripting and Malicious Modification
Affected Plugin: Page Builder: PageLayer – Drag and Drop website builder
Plugin Slug: pagelayer
Affected Versions: <= 1.1.1
CVE ID: CVE-2020-35947
CVSS Score: 7.4 (High)
CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:L
Fully Patched Version: 1.1.2

PageLayer is a very easy to use WordPress page builder plugin that claims to work with nearly all themes on the market and in the WordPress repository. It provides extended customization of pages through the use of widgets that can add page elements like buttons, tables, excerpts, products and more.

We discovered that nearly all of the AJAX action endpoints in this plugin failed to include permission checks. This meant that these actions could be executed by anyone authenticated on the site, including subscriber-level users. As standard, these AJAX endpoints only checked to see if a request was coming from /wp-admin through an authenticated session and did not check the capabilities of the user sending the request.

There were nonce checks in use in all of these functions, but nonces can be easily compromised if incorrectly implemented – for example, if a usable nonce is displayed within the source code of the site's output. Unfortunately for the PageLayer plugin, this is precisely what happened. A usable nonce was visible in the header section of the source code of any page that had previously been edited using the PageLayer plugin. Any site visitor could find this nonce, whether they were logged in or not, allowing any unauthenticated user the ability to obtain a legitimate nonce for the plugin's AJAX actions.


PageLayer nonce obtainable from page source.

Using a single nonce as the mechanism for authorization control caused various security issues in the functionalities of the page builder due to this nonce being so easily obtainable.

WordPress nonces should never be used as a means of authorization as they can easily be compromised if implemented improperly or if a loophole is found. WordPress nonces are designed to be used for CSRF protection, not authorization control. Implementing capability checks in conjunction with CSRF protection on sensitive functions for full verification provides protection to ensure a request is coming from an authorized user.

The Impact

As previously mentioned, several AJAX functions were affected, causing a large variety of potential impacts. A few of the most impactful actions were `wp_ajax_pagelayer_save_content`, `wp_ajax_pagelayer_update_site_title`, and `wp_ajax_pagelayer_save_template`.

```
122 | add_action('wp_ajax_pagelayer_save_content', 'pagelayer_save_content');
```

```
314 | add_action('wp_ajax_pagelayer_update_site_title', 'pagelayer_update_site_title');
```

```
940 | add_action('wp_ajax_pagelayer_save_template', 'pagelayer_save_template');
```

The `pagelayer_save_content` function is used to save a page's data through the page builder. The lack of permission checks on this function allowed authenticated users, regardless of permissions, the ability to change any data on a page edited with PageLayer.

```
function pagelayer_save_content(){

    $postID = (int) $_GET['postID'];

    if(empty($postID)){
        $msg['error'] =  __pl('invalid_post_id');
    }
```

An attacker could wipe the pages completely or inject any content they would like on the site's pages and posts. In addition, a few widgets allowed Javascript to be injected, including the "Button" widget. There is no sanitization on the "Button" widget's text, which allows for malicious Javascript to be used as a text. This Javascript would execute once any user browsed to a page containing that button.

PageLayer button with alert JS injected.

The `pagelayer_update_site_title` function is used to update a site's title. The lack of permission checks on this function allowed authenticated users the ability to change a site title to any title of their choosing. Though less detrimental, this could still affect your sites search engine ranking if unnoticed for an extended period of time.

```
function pagelayer_update_site_title(){
    global $wpdb;

    // Some AJAX security
    check_ajax_referer('pagelayer_ajax', 'pagelayer_nonce');

    $site_title = $_POST['site_title'];

    update_option('blogname', $site_title);

    $wpdb->query("UPDATE `sm_sitemeta`
                SET meta_value = '".$site_title."'
                WHERE meta_key = 'site_name'");
    wp_die();
}
```

The `pagelayer_save_template` function is used to save PageLayer templates for the PageLayer Theme Builder. The lack of permission checks on this function allowed authenticated users the ability to create new PageLayer templates that were saved as new posts.

```
function pagelayer_save_template() {

    // Some AJAX security
    check_ajax_referer('pagelayer_ajax', 'pagelayer_nonce');

    $done = [];

    $post_id = (int) $_GET['postID'];

    // We need to create the post
    if(empty($post_id)){

        // Get the template type
        if(empty($_POST['pagelayer_template_type'])){
            $done['error'] = __pl('temp_error_type');
            pagelayer_json_output($done);
        }

        $ret = wp_insert_post([
            'post_title' => $_POST['pagelayer_lib_title'],
            'post_type' => 'pagelayer-template',
            'post_status' => 'publish',
            'comment_status' => 'closed',
            'ping_status' => 'closed'
        ]);
```

Though this function was intended to be used in the PRO version of the plugin, the function could still be executed in the free version, affecting all 200,000+ users of the PageLayer plugin. An attacker could create a new template, which created a new page on the site, and inject malicious Javascript in the same way they could with the `pagelayer_save_content` function.

Malicious Javascript can be used to inject new administrative users, redirect site visitors, and even exploit a site's user's browser to compromise their computer.

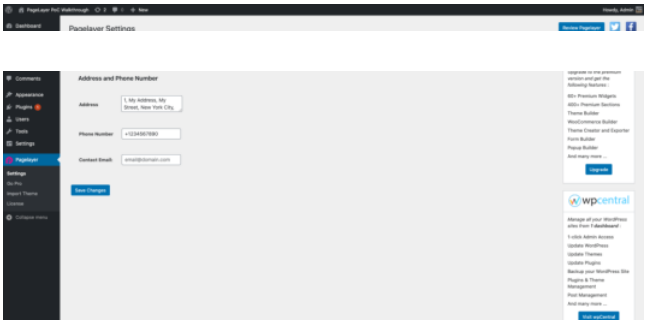
The Patch

In the latest version of the plugin, the developers implemented permissions checks on all of the sensitive functions that could make changes to a site, and reconfigured the plugin to create separate nonces for the public and administrative areas of a WordPress site.

```
// Are you allowed to edit ?
if(!pagelayer_user_can_edit($postID)){
    $msg['error'][] =  __pl('no_permission');
    pagelayer_json_output($msg);
}
```

Description: Cross-Site Request Forgery to Stored Cross-Site Scripting
Affected Plugin: Page Builder: PageLayer – Drag and Drop website builder
Plugin Slug: pagelayer
Affected Versions: <= 1.1.1
CVE ID: CVE-2020-35944
CVSS Score: 8.8 (High)
CVSS Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
Fully Patched Version: 1.1.2

The PageLayer plugin registers a settings area where configuration changes can be made. This includes functionality such as where the editor is enabled, basic content settings, basic information configurations, and more.

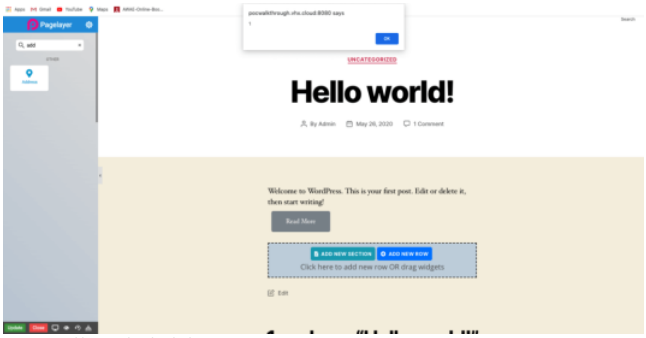


PageLayer settings area.

The settings update function used a capability check to verify that a user attempting to make any changes had the appropriate permissions. However, there was no CSRF protection to verify the legitimacy of any request attempting to update a site's settings. This made it possible for attackers to trick an administrator into sending a request to update any of the PageLayer settings.

```
function pagelayer_settings_page(){

    $option_name = 'pl_gen_setting' ;
    $new_value = '';

    if(isset($_REQUEST['pl_gen_setting'])){
        $new_value = $_REQUEST['pl_gen_setting'];

        if ( get_option( $option_name ) !== false ) {

            // The option already exists, so we just update it.
            update_option( $option_name, $new_value );
```

The "Information" tab in the settings area provides site owners with a way to set a default address, telephone number, and contact email address that are displayed whenever the corresponding widgets were used on a page. There was no sanitization on the address or telephone number settings, and due to the administrator's capability to use `unfiltered_html`, Javascript could be injected into these settings.



PageLayer Address updated with alert JS.

The Impact

This allowed attackers the ability to inject malicious scripts while exploiting the CSRF vulnerability in the settings. If the widget was already enabled, any injected malicious scripts would execute whenever someone browsed to a page containing that widget. If the widget was not yet enabled, the malicious scripts could be executed once an administrator started editing and inserting the widget into a page. As always, these scripts can do things like create a new administrative account and redirect users to malicious sites.

The Patch

In the patched version of the plugin, the developers implemented CSRF protection consisting of a WordPress nonce and verification of that nonce when updating settings.

```
if(isset($_REQUEST['submit'])){
    check_admin_referer('pagelayer-options');
}
```

PoC Walkthrough: pagelayer_save_content

Disclosure Timeline

April 24, 2020 to April 30, 2020 – Initial discovery of minor security flaw and deeper security analysis of plugin.
April 30, 2020 – Firewall rule was released for Wordfence Premium customers. We made our initial contact attempt with the plugin's development team.
May 1, 2020 – The plugin's development team confirms appropriate inbox for handling discussion. We provide full disclosure.
May 2, 2020 – Developer acknowledges receipt and confirms that they are beginning to work on fixes. An update is released the same day.
May 4, 2020 – We analyze the fixes and discover a few security issues left unpatched and responsibly disclose these issues to the developer.
May 6, 2020 – Developer releases the final sufficient patch.
May 30, 2020 – Free Wordfence users receive firewall rule.

Conclusion

In today's post, we detailed several flaws related to unprotected AJAX actions and nonce disclosure that allowed for attackers to make several malicious modifications to a site's pages and posts in addition to providing attackers with the ability to inject malicious Javascript. These flaws have been fully patched in version 1.1.2. We recommend that users immediately update to the latest version available, which is version 1.1.4 at the time of this publication.

Sites running Wordfence Premium have been protected from attacks against this vulnerability since April 30, 2020. Sites running the free version of Wordfence will recieve this firewall rule update on May 30, 2020. If you know a friend or colleague who is using this plugin on their site, we highly recommend forwarding this advisory to them to help keep their sites protected.

Comments

No Comments