


```

123 function pagelayer_save_content(){
124     ...
125
129     $postId = (int) $_GET['postId'];
130
131     if(empty($postId)){
132         $msg['error'] = _pl('invalid_post_id');
133     }
134 }

```

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.

```

315 function pagelayer_update_site_title(){
316     global $wpdb;
317     // Some AJAX security
318     check_ajax_referer('pagelayer_ajax', 'pagelayer_nonce');
319
320     $site_title = $_POST['site_title'];
321
322     update_option('blogname', $site_title);
323
324     $wpdb->query("UPDATE 'sm_sitemeta'
325                 SET meta_value = '". $site_title. "'
326                 WHERE meta_key = 'site_name'");
327
328     wp_die();
329 }

```

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.

```

941 function pagelayer_save_template() {
942     // Some AJAX security
943     check_ajax_referer('pagelayer_ajax', 'pagelayer_nonce');
944
945     $done = [];
946
947     $post_id = (int) $_GET['postId'];
948
949     // We need to create the post
950     if(empty($post_id)){
951         // Get the template type
952         if(empty($_POST['pagelayer_template_type'])){
953             $done['error'] = _pl('temp_error_type');
954             pagelayer_json_output($done);
955         }
956
957         $ret = wp_insert_post([
958             'post_title' => $_POST['pagelayer_lib_title'],
959             'post_type' => 'pagelayer-template',
960             'post_status' => 'publish',
961             'comment_status' => 'closed',
962             'ping_status' => 'closed'
963         ]);
964     }
965 }

```

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.

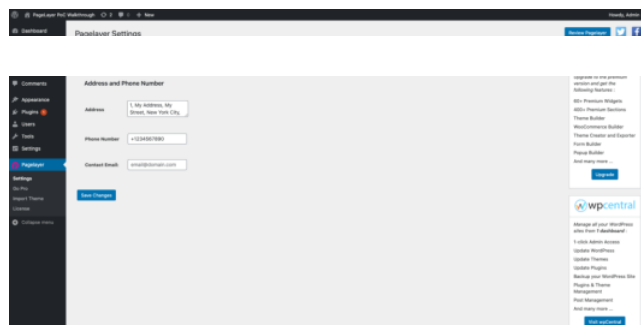
```

150 // Are you allowed to edit ?
151 if(!pagelayer_user_can_edit($postId)){
152     $msg['error'][] = _pl('no_permission');
153     pagelayer_json_output($msg);
154 }

```

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.



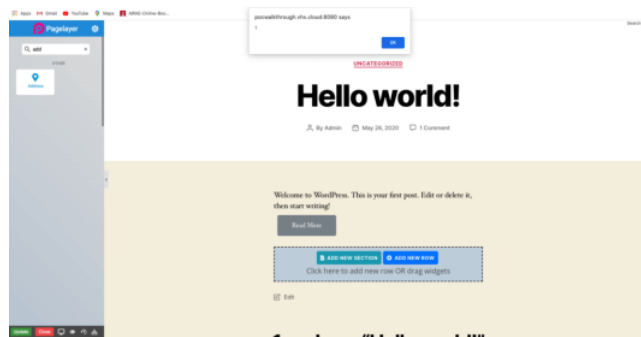
PageLayer settings area.

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.

```
156 function pagelayer_settings_page(){
157     $option_name = 'pl_gen_setting' ;
158     $new_value = '';
159     if(isset($_REQUEST['pl_gen_setting'])){
160         $new_value = $_REQUEST['pl_gen_setting'];
161         if ( get_option( $option_name ) != false ) {
162             // The option already exists, so we just update it.
163             update_option( $option_name, $new_value );
164         }
165     }
166 }
```



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.

```
176 if(isset($_REQUEST['submit'])){
177     check_admin_referer('pagelayer-options');
178 }
```



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 receive 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.

Did you enjoy this post? Share it!

Comments

No Comments

Breaking WordPress Security Research in your inbox as it 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.

[Terms of Service](#)

[Privacy Policy](#)

[CCPA Privacy Notice](#)



Products

[Wordfence Free](#)
[Wordfence Premium](#)
[Wordfence Care](#)
[Wordfence Response](#)
[Wordfence Central](#)

Support

[Documentation](#)
[Learning Center](#)
[Free Support](#)
[Premium Support](#)

News

[Blog](#)
[In The News](#)
[Vulnerability Advisories](#)

About

[About Wordfence](#)
[Careers](#)
[Contact](#)
[Security](#)
[CVE Request Form](#)

Stay Updated

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

☐ By checking this box I agree to the [terms of service](#) and [privacy policy](#).*

[SIGN UP](#)

© 2012-2022 Defiant Inc. All Rights Reserved