Ram Gall

December 10, 2020

Reflected XSS in PageLayer Plugin Affects Over 200,000 WordPress Sites

On November 4, 2020, the Wordfence Threat Intelligence team found two reflected Cross-Site Scripting (XSS) vulnerabilities in PageLayer, a WordPress plugin installed on over 200,000 sites. These vulnerabilities could lead to an attacker executing malicious Javascript in an administrator's browser, which could lead to takeover of a vulnerable WordPress site.

We contacted the plugin's publisher, Softaculous, on November 6, 2020, received a response over the weekend, and submitted the full disclosure on November 8, 2020. A patch was released the next day, November 9, 2020.

All sites running Wordfence, including Wordfence Premium customers as well as those still running the free version, are protected against this vulnerability by the Wordfence Firewall's built-in XSS protection.

Description: Multiple Reflected Cross-Site Scripting(XSS)
Affected Plugin: Page Builder: PageLayer – Drag and Drop website builder
Plugin Slug: pagelayer
Affected Versions: < 1.3.5
CVE ID: Will be updated once identifier is supplied.
CVSS Score: 6.1 (Medium)
CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Fully Patched Version: 1.3.5

The PageLayer plugin includes a settings page that allows site designers to select default font and color options to be used by the page builder. It accepts these options via various `$_POST` parameters. For example `body[font-size]` or `h3[font-size]` could be used to set the font size for `body` or `h3` tags, and `color[background]` could be used to set the background color.

The function used to modify these settings, `pagelayer_website_settings`, contained a capability check and a nonce to ensure that only valid, authorized requests could make changes.

A request submitted without the `submit` parameter would not save these changes, and instead continue to the `pagelayer_website_settings_t` function used to display forms on the settings page. Unfortunately, this function also called two other functions that accepted user input in order to display settings, and this is where an attacker could inject malicious JavaScript that could lead to takeover of a WordPress site.

XSS in the font-size parameter

```
400 function pagelayer_website_font_settings($prefix){
401
402     global $pagelayer, $pl_error;
403
404     if(!empty($_POST)){
405         $vals = $_POST;
406     }else{
407         $vals = $pagelayer->settings;
408     }
409
410     ?>
411
412     <table>
413         <tr>
414             <th scope="row"><?php echo __pl('font_family'); ?></th>
415             <td>
416                 <label>
417                     <select name="<?php echo $prefix;?>[font-family]">
418                         <?php
419                             foreach($pagelayer->fonts as $k => $font){
420                                 echo '<option value="'.$font.'" '.($vals[$prefix]['font-family'] == $font ? 'selected
421                             }
422                         ?>
423                     </select>
424                 </label>
425             </td>
426         </tr>
427
428         <tr>
429             <th scope="row"><?php echo __pl('font_size'); ?></th>
430             <td>
431                 <label>
432                     <select class="pagelayer-show-custom" onchange="pagelayer_handle_custom(this)">
433                         <option value="" <?php echo (empty($vals[$prefix]['font-size']) ? 'selected="seleted"' : '');
434                         <option value="custom" <?php echo (!empty($vals[$prefix]['font-size']) ? 'selected="seleted"'
435                     </select>
436                     <input type="number" name="<?php echo $prefix;?>[font-size]" <?php echo (!empty($vals[$prefix]['f
437                 </label>
438             </td>
439         </tr>
440
```

The `pagelayer_website_font_settings` function accepts input from the `$_POST` parameter and echoes out the `$prefix['font-size']` subparameter each time the function is run, where `$prefix` is a different tag such as `body` or `h3`. As such, if an attacker could trick an administrator into clicking a link that submitted a POST request containing, for instance, a `body[font-size]` parameter set to a malicious script, that script would be executed in the administrator's browser.

XSS in color settings

```
383     global $pagelayer, $pl_error;
384
385     $val = !empty($_POST) ? @$_POST['color'][$field] : @$pagelayer->settings['color'][$field];
386
387     echo '
388     <table>
389         <tr>
390             <th scope="row">'.$text.'</th>
391             <td>
392                 <a href="#" class="pagelayer-show-vanilla"><div class="pagelayer-color-div pagelayer-color-none"></di
393             </td>
394         </tr>
395     </table>';
396
397 }
```

The `pagelayer_website_color` function also accepts input from the `$_POST` parameter, but echoes out the `color[$value]` subparameter each time the function is run, where `$value` is a CSS selector such as `background` or `link`. If an attacker could trick an administrator into clicking a link that submitted a POST request containing a `color` sub parameter such as `color[background]` set to a malicious script, that script would be executed in the administrator's browser.

We've previously discussed reflected XSS and how it can be just as dangerous as stored Cross-Site Scripting (XSS), especially when deployed against a site administrator. Exploits targeting site administrators could be used to take over a site by creating malicious administrative accounts or by embedding backdoors in theme files. One of the reasons we recommend that plugin developers escape everything that can be modified by user input is that even seemingly innocuous parameters, such as font-size, can be used to output malicious JavaScript.

Timeline

November 4, 2020 – Wordfence Threat Intelligence discovers reflected XSS vulnerabilities in the PageLayer plugin.
November 6, 2020 – We finish analyzing the vulnerabilities and initiate the disclosure process.
November 8, 2020 – We send our full disclosure to the plugin publisher.
November 9, 2020 – Softaculous releases a patch.

Conclusion

In this article, we discussed two reflected XSS vulnerabilities in the administrative panel of the PageLayer plugin. These vulnerabilities have been fully patched in version 1.3.5 and we strongly recommend all users update to the latest version, 1.3.8 as of this writing. All Wordfence users, including sites running Wordfence Premium as well as those still using the free version, are protected by the Wordfence Firewall's built-in XSS protection.

Special Thanks to Softaculous, the makers of the PageLayer plugin, for their rapid response in correcting this issue.

Comments

2 Comments

jeff *
December 11, 2020
7:37 am

Can you describe how they fixed this?

Ram Gall *
December 14, 2020
6:04 am

Hi jeff,

They escaped the output on these as well as enforcing a nonce check on all $_POST requests to the menu page rather than only on requests to update the settings.