



Chloe Chamberland

October 7, 2020

Vulnerability Exposes Over 4 Million Sites Using WPBakery

On July 27th, our Threat Intelligence team discovered a vulnerability in [WPBakery](#), a WordPress plugin installed on over 4.3 million sites. This flaw made it possible for authenticated attackers with contributor-level or above permissions to inject malicious JavaScript in posts.

We initially reached out to the plugin's team on July 28, 2020 through their support forum. After receiving confirmation of the appropriate support channel, we disclosed the full details on July 29, 2020. They confirmed the vulnerability and reported that their development team had begun working on a fix on July 31, 2020. After a long period of correspondence with the plugin development team, and a number of insufficient patches, a final sufficient patch was released on September 24, 2020.

We highly recommend updating to the latest version, 6.4.1 as of today, immediately. While doing so, we also recommend verifying that you do not have any untrusted contributor or author user accounts on your WordPress site.

Wordfence Premium users have been protected against exploits targeting these vulnerabilities since July 28, 2020. Wordfence free users received the same protection on August 28, 2020.

Description: Authenticated Stored Cross-Site Scripting (XSS)

Affected Plugin: WPBakery

Plugin Slug: js_composer

Affected Versions: <= 6.4

CVE ID: [CVE-2020-28650](#)

CVSS Score: 6.4 Medium

CVSS Vector: [CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C/L/H/L/A/N](#)

Fully Patched Version: 6.4.1

WPBakery page builder is the most popular page builder for WordPress. It is a very easy to use tool that allows site owners to create custom pages using drag and drop capabilities.

Unfortunately, the plugin was designed with a flaw that could give users with contributor and author level roles the ability to inject malicious JavaScript into pages and posts. This flaw also gave these users the ability to edit other users' posts. The plugin explicitly disabled any default post HTML filtering checks in the `saveAjaxFe` function using `kSES_remove_filters()`. This meant that any user with access to the WPBakery builder could inject HTML and JavaScript anywhere in a post using the page builder.

```
32 public function saveAjaxFe() {
33     vc_user_access()->checkAdminNonce()->validateDie()->wpAny( 'edit_posts', 'edit_pages' )->validateDie();
34
35     $post_id = intval( vc_post_param( 'post_id' ) );
36     if ( $post_id > 0 ) {
37         ob_start();
38
39         // Update post_content, title and etc.
40         // post_title
41         // content
42         // post_status
43         if ( vc_post_param( 'content' ) ) {
44             $post = get_post( $post_id );
45             $post->post_content = stripslashes( vc_post_param( 'content' ) );
46             $post->status = vc_post_param( 'post_status' );
47             $post->title = vc_post_param( 'post_title' );
48             if ( null !== $post->title ) {
49                 $post->post_title = $post->title;
50             }
51             kSES_remove_filters();
52             remove_filter( 'content_save_pre', 'balanceTags', 50 );
```

Furthermore, while WPBakery only intended pages that were built with the WPBakery page builder to be editable via the builder, users could access the editor by supplying the correct parameters and values for any post. This could be classified as a general bug as well as a security issue, and is what made it possible for contributors and editors to use the `wp_ajax_vc_save` AJAX action and corresponding `saveAjaxFe` function to inject malicious JavaScript on their own posts as well as other users' posts.

The plugin also had custom onclick functionality for buttons. This made it possible for an attacker to inject malicious JavaScript in a button that would execute on a click of the button. Furthermore, contributor and author level users were able to use the `vc_raw_js`, `vc_raw_html`, and button using `custom_onclick` shortcodes to add malicious JavaScript to posts.

All of these meant that a user with contributor-level access could inject scripts in posts that would later execute once someone accessed the page or clicked a button, using various different methods. As contributor-level users require approval before publishing, it is highly likely that an administrator would view a page containing malicious JavaScript created by an attacker with contributor-level access. By executing malicious JavaScript in the administrator's browser, it would be possible for an attacker to create a new malicious administrative user or inject a backdoor, among many other things.

In the latest version of WPBakery, lower level users no longer have `unfiltered_html` capabilities by default, however, administrators can grant that permission if they wish to. In addition, users without the appropriate privileges can no longer edit other users' posts, access the page builder unless permitted, or use shortcodes that could allow the injection of malicious JavaScript.

Dual Account Control

One strategy to keep your site protected from Cross-Site Scripting attacks against higher-privileged accounts is to use dual accounts. Dual account control uses two accounts for any user that may require administrative capability. This can be done by using one user account with administrative capabilities for admin-related tasks like adding new users and plugins and another user account with editor capabilities used to review and approve author and contributor posts.

Doing so will limit the impact that a Cross-Site Scripting vulnerability may have. When you access a page as a site administrator you will notice a JavaScript that an attacker injects can use administrative web functions. By adding a new

checking, and ensuring no more checks of user level access, the injection attempt seems to miss, so an attacker can't successfully add new admin accounts or edit themes through an Editor account.

Especially in cases where many users can access authenticated actions, we recommend using an administrative user account only when you need to perform administrative functions on your site.

Disclosure Timeline

July 27, 2020 – Initial discovery of the vulnerability. We develop a firewall rule and move it into the testing phase.

July 28, 2020 – The firewall rule is sufficiently tested and released to premium users. We make our initial outreach to the WPBakery plugin team.

July 29, 2020 – The WPBakery team responds confirming the appropriate inbox and we send over full disclosure details.

August 21, 2020 – After some follow-up an initial patch is released.

August 26, 2020 – We let the WPBakery team know that there are some additional minor problems missed that require resolution.

August 28, 2020 – Wordfence free users receive the firewall rule.

September 2, 2020 – We follow up to see if the WPBakery team received our last email.

September 9, 2020 – The WPBakery team confirms they received our email and are working on getting an additional patch released.

September 11, 2020 – The WPBakery team releases an additional patch that is not fully sufficient.

September 11 to 23, 2020 – We work together more closely to get an adequate patch out.

September 24, 2020 – Final sufficient patch released in version 6.4.1.

Conclusion

In today's post, we detailed a flaw in the WPBakery Plugin that provided authenticated users with the ability to inject malicious JavaScript into posts using the WPBakery Page builder. Along with that, we provided some insight on how you can protect yourself against Contributor and Author level vulnerabilities. This flaw has been fully patched in version 6.4.1. We recommend that users immediately update to the latest version available, which is version 6.4.1 at the time of this publication.

As WPBakery is a premium plugin often included as a page builder with numerous premium themes, you may need to double check that any updates are available to you with your theme purchase. Verifying the plugin version number in your plugins dashboard should alert you to the version installed on your site.


Sites using [Wordfence Premium](#) have been protected against attacks attempting to exploit this vulnerability since July 28, 2020. Sites still using the free version of Wordfence received the same protection on August 28, 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 as this is a significant security update.


Did you enjoy this post? [Share it!](#)

Comments

22 Comments

 **Wendy ***
October 7, 2020
10:17 am


Hi there,
My Wordpress Updates page is telling me that compatibility with WP 5.5.1 is unknown. Is that accurate?

 **Chloe Chamberland ***
October 8, 2020
8:18 am


Hi Wendy,

It does look like the WordPress Updates page says that compatibility with WP 5.5.1 is unknown for this update. I was able to update with no issues, however, I would recommend reaching out to the WPBakery Support team to verify that there won't be any issues. It is possible they just missed updating the flag for this release.


Chloe

 **Lisa Smith ***
October 7, 2020
10:30 am

So I'm confused, is it the Page Builder plugin (that has completely different version number) or what?

 **Lisa Smith ***
October 7, 2020
10:38 am


APPARENTLY THE WORDFENCE EMAIL had a typo in the version number ... it read 4.6.1 instead of 6.4.1.

 **Chloe Chamberland ***
October 7, 2020
10:46 am


Hi Lisa,

I am so sorry about any confusion! You are correct, there is a typo in the version number in the email. This does affect the WPBakery Page Builder and the version you should update to is 6.4.1. Once again, I apologize for the mistake.

Chloe

 **Tony Papas ***
October 7, 2020
3:22 pm


This plugin has been giving me a little trouble lately however my version was purchased with the template however since then the plugin have been upgraded to 6.2.0 and its not showing or upgrading any further. can you provide us with some assistance to find this upgrade please and is it free ?

 **Chloe Chamberland ***
October 8, 2020
7:09 am

Hi Tony,

If WPBakery came with a theme, I would recommend reaching out to the company that developed your theme to ask them for an updated copy of WPBakery. In the meantime, Wordfence will keep you protected.

Chloe

 **spwebmaster ***
October 7, 2020
5:02 pm

Hello,
Could you please WPBakery Visual Composer 6.4.0 to 6.4.4 and those patches were fixed with one click?

<https://snipboard.io/qaVKrB.jpg>

If you don't quite know the answer, maybe you can point me in the right direction?
Thanks so much!



Chloe Chamberland *

October 8, 2020
7:12 am

Hi there!

Based on the error, it looks like you do not have a license key registered in the WPBakery plugin. If you purchased the plugin directly from them, then I recommend obtaining your WPBakery license key from your account with them. If WPBakery came with a theme, I would recommend reaching out to the company that developed your theme to ask for an updated copy of the plugin.

Chloe



Kenneth *

October 7, 2020
6:31 pm

I'm using a really old version (4.12) that came with a wordpress theme i bought. Back when it was called WPBakery Visual Composer. I tried to look for the functions saveAjaxFe and kses_remove_filters, but couldn't find them. Should I buy the latest version?



Chloe Chamberland *

October 8, 2020
7:29 am

Hi Kenneth,

Those functions reside in the class-vc-post-admin.php file if that helps with your search. If you do not see them there, then I would say you are not vulnerable and are not obligated to buy the latest version. It is possible that this vulnerability was introduced during the switchover from the visual composer branding, though we do not have copies of old versioning to confirm. With that being said, I never recommend remaining on out of date software so I would definitely still recommend looking into upgrading to the latest version available.

Chloe



Mike *

October 8, 2020
3:53 pm

Not in version 5.2 either.



Huy Hoa *

October 7, 2020
9:22 pm

Hi Chloe,
First, thanks for detail information.
I'm using Wordfence free version for my personal website. About this vulnerability, it will effect to site with multiple author and have at least Author permission or higher, right?
If my site have only 1 admin account so it's safe?
I'm waiting for my theme provider to update the latest version which include Wp Bakery 6.4.1



Chloe Chamberland *

October 8, 2020
7:18 am

Hi Huy,

That is correct, this vulnerability can only be exploited by those with contributor and author permissions. If you only have a single administrator on your site, then your site will be fine while waiting for the update from your theme provider.

Chloe



Pps *

October 8, 2020
2:25 pm

And if I have only 2 or 3 user with admin level? Can this bug affect my site?



Chloe Chamberland *

October 13, 2020
7:27 am

Hi there,

Your site should be fine if you only have 2 or 3 admin accounts. Administrative users can add unfiltered HTML to posts by default so that is expected. This vulnerability can only affect you if you have contributor or author level users on your site since they should not be able to add unfiltered HTML to posts by default.

Chloe



Hristo *

October 7, 2020
11:20 pm

Hello,

One question - after update, is there any way to check if the site has been a victim of the vulnerability? And to remove potential XSS scripts?

Also, does the update fix potential XSS redirections or should this be done manually? How can potential exploits be located?

Thanks!



Chloe Chamberland *

October 8, 2020
7:39 am

Hi Hristo!

We do not have any evidence that suggests that this vulnerability was known about prior to it getting patched, nor do we have any evidence that suggests it is actively being exploited so your site should not have already been a victim of this vulnerability. However, with that being said, I would recommend running a Wordfence scan and following [this guide to clean a hacked site](#). If you would like to verify your site hasn't been compromised. More specifically, [this page would help you the most with finding any potential XSS](#).

The update would not fix any already injected XSS, however, we have no evidence to suggest this was, or is, being exploited.

I hope that helps!
Chloe



Jade *

October 8, 2020
2:41 am

Good Day

Does this affect the WPBakery Visual Composer Plugin as well?



October 8, 2020
7:56 am

Hi Jade,

If you are referring to the Visual Composer plugin available in the WordPress repository, then no that plugin is not affected. If you are referring to the old version of WPBakery which was referred to as Visual Composer, then unfortunately we can not say for certain as we do not have access to older versions of the plugin. Without knowing for certain, it is best to assume it is vulnerable and update as soon as possible.

I hope that helps clarify!
Chloe



Vistor •
October 12, 2020
7:02 pm

I am a little confused on this issue.

An authenticated attackers with contributor-level or above permissions can inject malicious JavaScript in posts.

Doesn't a contributor already have the ability to add a JS script to any page?



Chloe Chamberland •
October 13, 2020
7:38 am

Hi there,

Contributors do not have the ability to add JavaScript to posts, and should not unless explicitly granted by an administrator. In WordPress there is a permission called `unfiltered_html` which allows users to insert unfiltered HTML and JavaScript in posts and pages. This permission is only granted to administrator and editor level users, meaning only admins and editors should be able to add JavaScript and/or HTML to any page or post without any filtering. Contributors, subscribers, and authors do not have the `unfiltered_html` capability therefore any JS or HTML added into posts or pages gets sanitized. This is a security feature to prevent lower-level users from being able to XSS higher level users. WPBakery essentially disabled this protection on posts making it possible for lower-level users like contributors and authors to inject malicious JavaScript into posts and pages even though they should not have been able to without explicit permission from the site owner.

Hope that helps!
Chloe

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-6pm 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