UPDATED: 08.08.2022

Listen: Patchstack Weekly #35: SVG XSS Reported in Gutenberg

Robert Rowley
from **patchstack**

Welcome back to the Patchstack Weekly Security Update! This update is for week 32 of 2022.

It is August, and the Patchstack Alliance is growing. New security researchers have joined the alliance in the last month, and we are receiving some great reports serious security bugs in open source components affecting millions of websites (based on installation counts). Every one of these reports is handled professionally

their sites and protect their users before the information about the bug is made public.

This week there was a security bug that went public, but it was not handled by Patchstack Alliance. This new security bug is in the WordPress editor: Gutenberg and was handled through WordPress.org's HackerOne bug bounty program. There is no patch available for this bug, but don't panic, it is not a critically severe issue, it is at best a low severity issue (and possibly not even that). In this week's knowledge share I will share important details that will help you understand the low risk this now public vulnerability poses, and emphasize that the existence of a CVE is in itself not a sign of high risk - because severity matters too.

I will then cover the week's vulnerability news, covering 3 plugins (including Gutenberg) with security bugs, only 1 of the three has a patch available at this time.

SVG XSS reported in Gutenberg

CVE-2022-33994 is being looked into. The reported vulnerability was first published over the weekend and had a high severity rating but now it is currently "under review" according to NVD (the National Vulnerability Database), with no severity rating.

an SVG added via "insert URL" in Gutenberg editor is only dangerous if some very specific conditions are met, conditions any modern browser and properly configured web application protect users against.

Protections like: Same-origin policy. All major browsers implemented this protection years ago, and it addresses the exact risk this security bug presents. The purpose of same-origin policy is to prevent documents, media, or scripts from interacting with resources from other origins (an origin is a combination of domain name, protocol, and port). If the SVG file was uploaded directly to the website, it would have the same origin and pose a risk, but that is not what this CVE is about, so it is not a risk.

CORS (cross-origin resource sharing) headers are something I should mention too. CORS headers are managed by the web application and inform the browser what other origins (domains) may interact with resources for that application. A valuable tool to make same-origin policy work and still be secure across domains, because modern web applications may need to allow multiple origins to interact with one origin's resources being made available another. When adding new origins with a CORS header, developers should be careful and only allow trusted domains, a site would be in danger if the CORS headers allowed any origin to interact with resources on the page, such as if they used an asterisk *. Misconfigured CORS headers could happen, but this would be an extremely unsafe choice to make, so we can't assign risk to this CVE based on a misconfiguration because that misconfiguration would be a security risk in its own right.

The final and most important point to bring up is that Gutenberg's "Insert URL" feature utilizes `<img>` tags to add the remote SVG file. Modern browsers that load SVG files via an `<img>` tag, will not run any javascript in the SVG file, making yet another hurdle (also known as defense in depth) for attackers to overcome to perform a successful attack. As long as images added via Insert URL use the `<img>` tags, this CVE does not pose a risk.

Remotely loaded SVGs are not an immediate threat, but they are a security consideration. Many big players on the web (like Reddit, Facebook, and Google) err on the side of caution and do not allow users to embed remotely loaded SVGs because of the unlikely but possible risk. The big players know it is unlikely to bypass all of the protections in modern browsers like same origin policy, cross-origin resource sharing headers, and `<img>` tag limitations. But, the big players know a hypothetical vulnerability is worth protecting against.

It would be nice if this was addressed, but there is no simple solution. Meanwhile, there is little risk to loading remotely hosted SVGs via `<img>` tags thanks to protections found in modern browsers. Users may be confused why they are not allowed to upload an SVG to the website directly while they are allowed to add SVG images using the "Insert URL" button. But, I just shared with you the important difference between these two thing. Modern browsers have built in protections like same-origin policy, CORS headers, and `<img>` tag restrictions, which make adding SVG files via "Insert URL" a lot safer than allowing SVG files to be directly uploaded.

Vulnerability news

Gutenberg - Authenticated Stored Cross-Site Scripting

This security bug was the topic of this week's knowledge share. While there is no

your website's CORS headers to not do anything unsafe, but there are multiple layers of protections that exist before this vulnerability can pose a higher risk.

Advanced Custom Fields - Unauthenticated File Upload

The advanced custom fields pro and free versions patched a security bug that could have allowed unauthenticated file uploads. The researcher who found and reported this issue is not sharing exploit details at this time but does acknowledge users can only upload file types WordPress allows (images, media) and not PHP files. Site owners should still update soon before more details get out there.

Duplicator - Unauthenticated System Information Disclosure

It is reported this security bug has no patch yet, but the duplicator's plugin should not be permanently installed on your websites as far as I can tell. The purpose of this plugin is to easily migrate or clone WordPress websites. I would recommend remove or disable any plugin when it is not in use. For duplicator, once you have

Thanks and appreciation

This week's thanks goes out to the developers of Advanced Custom Fields. Great job addressing that security bug in both your free and pro product.

Further thanks go out to the Patchstack Alliance team members for all of their efforts in finding and responsibly disclosing security bugs in open source components. We just announced the bug bounty prize pools for June, and August is off to a great start!

I will be back next week with more security tips, tricks, opinions, and news on the Patchstack Weekly Security Update!

Start listening

Related Articles

View All >

PATCHSTACK WEEKLY

Patchstack Weekly #49: Hunting Open-Source Security Bugs with SAST.

LAST PATCH, WORDPRESS PLUGINS

PATCHSTACK WEEKLY

Patchstack Weekly #48: Dealing with End of Life and Unsupported Open Source Projects.

Start FREE

Patchstack

About us

Careers

Media kit

Articles & insight

Whitepaper 2021

Social

LinkedIn

Facebook

Twitter

hackuu