Jun 5, 20 2:42 pm

Elementor Page Builder 2.9.8 Stored XSS

Jun 5, 2020 | by: Sherif Koussa

At Software Secured, we routinely test our own infrastructure to make sure we practice what we preach. Having good security hygiene brings us one step closer to understanding our clients. An elementor page builder such as WordPress helps with our security.

The Software Secured website is powered by WordPress, as it's easy to use for the whole team and has regular security updates, as well as a wealth of plugins that help ease the creation of content. From Wikipedia:

WordPress is used by more than 60 million websites, including 33.6% of the top 10 million websites as of April 2019, WordPress is one of the most popular content management system solutions in use.

During a routine test of our website, we decided to review our installed WordPress plugins and the existence of public vulnerabilities for those plugins. These third-party components pose potential risk that we've written about before. The first step was gaining a list of active plugins and their versions to compare with known vulnerabilities. We listed our plugins, checked our versions and found they were all up to date without vulnerabilities. We decided to dig deeper.

Elementor caught our attention, as a potential test target, as it's a powerful plugin that has an excellent feature set which provided a large scope for us to test. From the Elementor website:

A page builder that delivers high-end page designs and advanced capabilities, never before seen on WordPress.

There have been interesting vulnerabilities in the Elementor editor in the past:

https://wpvulndb.com/plugins/elementor

Most of the issues consist of cross-site scripting (XSS) vulnerabilities, which are a great bug class to search for, as it's rarely fixed across the entire web application and has serious impact. From OWASP:

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites.

We have other posts explaining more about XSS here and here.

The following details two different XSS vectors that we found while reviewing the Elementor page builder plugin.

XSS #1

Discovery:

We created a draft post and opened it in the Elementor editor. Elementor uses a drag-and-drop UI, where users pick UI widgets that they would like in their post. We started by looking at an image widget.

The Elementor image widget allows the use of a custom link, so that when a user clicks the image, they are navigated elsewhere on the site. This has potential for XSS!

Exploitation:

Custom links should ring alarm bells for security testers as links with the `javascript:` protocol are susceptible to XSS. Browsers will execute arbitrary JavaScript when used with a `javascript:` protocol, like the following:

```
javascript:alert('XSS')
```

We placed this into the custom link field and previewed the page. The results on the page looked like:

When the image is clicked, the XSS payload fires.

The above proof-of-concept was viable for all UI widgets that allowed the use of a custom URL.

XSS #2

Discovery

We continued testing with the image widget, and jumped to the advanced tab, as this is usually a good spot to look for extra configuration.

About the Author

Sherif Koussa

Sherif Koussa is OWASP Ottawa Chapter Co-Leader, Software Developer, Hacker, and founder and CEO of Software Secured and Reshift. In addition to contributing to OWASP Ottawa for over 14 years, Sherif contributed to WebGoat, and OWASP Cheat Sheets. Sherif also helped the SANS and GIAC organizations launch their GSSP-Java and GSSP-NET exams and contributed to a few of their courses. After switching from the software development field to the security field, Sherif took on the mission of supporting developers shifting security left, and ship more secure code organically.

Share This Post

The advanced tab has an option for custom attributes, which has XSS written all over it.

Related Post

Our first test looked like:

```
xss|xss
onload|alert()
onclick|alert()
onmouseover|alert()
```

As we didn’t know what we were dealing with yet. Here’s what we saw once we previewed the page:

Jul 29, 2022 by Mozart Data

How Start-Ups Can Build a Data-Driven Culture

Read more

The `xss|xss` row was added as an attribute, but the rest were all stripped. How do we find the correct HTML attributes that will be left unsanitized?

In comes the ever valuable PortSwigger XSS cheat sheet.

This payload caught our eye, as it affects all browsers and works on `div` elements:

Exploitation:

The trouble was that it required adding an animation style to the page and applying the style to our `div` element. Of course, Elementor has a solution for adding custom CSS to an element, and we could include the extra style attribute the same way we were including our other attributes:

We previewed our page and success!

Jun 27, 2022 by Martin Cozzi

What is a SOC 2 Report and Why Are Your Clients Asking For It?

Read more

This time, our attributes were not stripped or modified by the server, and our payload is returned to the browser, where it executes when the page loads.

The above proof-of-concept was viable for all UI widgets that Elementor supports, as they all support custom CSS and attributes.

We promptly wrote up both issues, did a happy dance, and then privately disclosed the issue to plugins@wordpress.org. We applied for a CVE to help the community track the above two vulnerabilities, and were given CVE-2020-13864. We offered to test any fixes as they come out, which lead to the next portion.

Bypass for XSS #1

Exploitation:

Once XSS #1 was fixed by the Elementor team, we re-opened our PoC draft blog post to validate the fix.

By utilizing the PortSwigger cheat sheet again, we were able to find a bypass for the initial fix. In particular we found that the following two payloads worked:

Nov 14, 2022 by Omkar Hiremath

NIST SP 800-115 and Penetration Testing

Read more

We reported our bypass and waited for a fix. The bypass was given the following CVE-2020-13865 to help the community track the vulnerability.

Impact:

A low privileged author user is able to launch XSS attacks against admin and unauthenticated victims, which can potentially lead to privilege escalation or malware delivery.

Recommendations:

Update to the latest version of the Elementor plugin.

Conclusion:

As our readers know, our team gets pretty excited about deep diving to find vulnerabilities like this one. We like helping companies reduce their risk and educating other developers and pentesters on what we learn!

If you're looking for security training or want to explore how this type of manual pentesting can make an impact on your organization's security culture, click the link below to get the conversation started.

Timeline:

1. May 20th 2020 - Bug found and private disclosure to plugins@wordpress.com
2. May 21st 2020 - plugins@wordpress.com acknowledge receipt of the disclosure
3. May 24th 2020 - Elementor fixed the issue in version 2.9.9
4. May 28th 2020 - Bypass submitted for the above fix
5. May 28th 2020 - plugins@wordpress.com acknowledge receipt of the bypass
6. June 1st 2020 - Elementor fixed the bypass in version 2.9.10
7. June 5th 2020 - CVE-2020-13864 and CVE-2020-13865 provisioned
8. June 5th 2020 – Blog post released

References:

1. https://en.wikipedia.org/wiki/WordPress
2. https://owasp.org/www-community/attacks/xss/
3. https://wordpress.org/plugins/elementor/
4. https://elementor.com/pro/changelog/
5. https://portswigger.net/web-security/cross-site-scripting/cheat-sheet
6. https://wpvulndb.com/plugins/elementor
7. https://www.softwaresecured.com//the-rise-of-javascript-xss-and-practical-mitigation-techniques
8. https://www.softwaresecured.com//jetbrains-teamcity-reflected-xss/
9. https://www.softwaresecured.com//13-tools-for-checking-the-security-risk-of-open-source-dependencies/
10. https://cve.mitre.org/
11. https://wpvulndb.com/vulnerabilities/10256

Was this post helpful? Let us know if you liked the post. That's the only way we can improve.

Yes

No