






SECURITY

Critical WebP bug: many apps, not just browsers, under threat

The heap buffer overflow (CVE-2023-4863) vulnerability in the WebP Codec is being actively exploited in the wild.

 Alex Ivanovs  September 13, 2023  [Reader Disclosure](#)

A significant vulnerability in the [WebP Codec](#) has been unearthed, prompting major browser vendors, including [Google](#) and [Mozilla](#), to expedite the release of updates to address the issue.

Update (9/13/2023): So far the Web Browsers that have confirmed a fix and released an update include: Google Chrome[[1](#)], Mozilla Firefox[[2](#)], Brave[[3](#)], Microsoft Edge[[4](#)], and Tor Browser[[5](#)]. If your browser of choice is using Chromium then expect an update to already be rolled out or will be done shortly.

⚠ Important: Let me make it perfectly clear that this vulnerability doesn't just affect web browsers, **it affects any software that uses the libwebp library**. This includes Electron-based applications, for example - Signal. Electron [patched the vulnerability yesterday](#). Also, software like [Honeyview](#) (from Bandisoft) released an update to fix the issue. CVE-2023-4863 was falsely marked as Chrome-only by Mitre and other organizations that track CVE's and 100% of media reported this issue as "Chrome only", when it's not.



👉 **Who uses libwebp?** There are a lot of applications that use libwebp to render WebP images, I already mentioned a few of them, but some of the others that I know include: **Affinity** (the design software), **Gimp**, **Inkscape**, **LibreOffice**, **Telegram**, **Thunderbird** (now patched), **ffmpeg**, and many, many **Android applications** as well as **cross-platform apps built with Flutter**.

Update (9/14/2023): 1Password for Mac have [released an update](#) to address the issue. 1Password (like many others) is an application built with Electron, and until all these apps upgrade to the latest version - they are considered vulnerable based on the severity of the bug.

Update (9/15/2023): Okay, so, I thought I would give an update as I have been getting a lot of emails about this, and I can't spend so much time trying to answer each one individually. I know that Telegram Desktop [made an update](#) and I have seen Ubuntu, Debian, SUSE and other Linux platforms also actively updating their libwebp versions. I also know that software like Obsidian is [going to bump their Electron version](#) to address the bug. But I don't know the full scope of this because there isn't a catalog of apps to browse to see who is using the WebP Codec or who isn't.

And one more thing.

How come Mitre marked this as Chrome-only?

I want to address this also because I don't want to leave an impression that Mitre made a mistake in assigning this CVE, what they should have done is make it clearer that the issue was broader than just Chromium. The way that CVE publishing works is that each issue (in this case a 0day in WebP Codec) is listed individually for each software, and the same CVE is assigned independently when the next software is found to have this bug.

If you're interested, you can [read this report from Citizen Lab](#) (September 7) on a zero-click exploit they found in the wild. Subsequently, Apple made an update (September 7) to their ImageIO library and called it a "buffer overflow" (I linked it further down below) but Apple never assigned it a CVE, instead

they disclosed it with Google first who then issued the fix (September 11).

This newly identified vulnerability, designated as **CVE-2023-4863**, pertains to a heap buffer overflow in the WebP image format. Google Chrome and Mozilla Firefox, among other browsers, use WebP for its efficient image compression capabilities. A malicious exploitation of this flaw could potentially jeopardize the security of millions of internet users.

What is a 'heap buffer overflow' and how bad is it?

I'm going to give the most beginner-friendly explanation I can to make this as understandable as possible:

Imagine you have a shelf that can only hold 5 books. This shelf is meant exactly for those 5 books, and no more. Now, what happens if you try to forcefully fit a 6th book? The shelf might break or the 6th book might push out another book from the shelf, right? That's the essence of "overflow" – trying to put more into something than it was designed to hold.

Computers use an area of memory called the "heap" to store certain kinds of data. If a program doesn't manage this memory well and tries to stuff more data into a "heap buffer" (like our shelf) than it can handle, that's called a heap buffer overflow.

Going back to our shelf analogy, imagine if you had a mischievous friend. Every time you tried to fit too many books on the shelf, your friend could decide which book gets pushed out. Maybe they replace your favorite novel with a book you hate. In computer terms, the data that overflows can overwrite other important data or instructions in a way that's beneficial to an attacker.

If someone knows a program has a heap buffer overflow vulnerability, they might be able to send it specially crafted data that causes the program to behave in unexpected ways. For instance, they could potentially run malicious code or gain unauthorized access to a system.

A codec is like a translator that helps your computer understand and display WebP images (a format like JPEG or PNG). If this codec has a heap buffer overflow, an attacker might be able to craft a malicious WebP image that, when viewed, exploits this vulnerability to harm your computer or steal information.

How bad is it? Very. If an attacker can exploit a heap buffer overflow, they might be able to take control of a system, steal data, or introduce malware. Protecting against such vulnerabilities is crucial.

What was the problem in the WebP library?

The root of the issue lies within the "**BuildHuffmanTable**" function which [was first introduced in 2014](#), the function is used to verify if the data is accurate. The vulnerability can occur when more memory is allocated if the table isn't sufficiently large for valid data. The commit that introduces the fix [can be seen here](#).

The original code optimized a Huffman decoder that uses a common technique: it reads several bits ahead to determine how many bits to consume and what symbol to decode. The older version utilized lookup tables for short symbols, while longer ones required a more complex graph traversal. The newer version streamlined this process by employing an array of lookup tables. Each entry in this table contains details about bits and values, and if the number of bits surpasses a certain limit, the value is interpreted differently.

The new version determined the maximum number of entries by counting symbols. However, because the Huffman tree comes from an untrusted source, situations could arise where the number of bits is excessively large. The VP8 Lossless allows up to 15 bits, which means the largest table can have many entries, more than it should. Interestingly, while there was a mode in the code to only calculate the table size, it was not used, and a fixed size was assumed, leading to potential overflows.

The reason behind these changes was to optimize the Huffman decoding step, a crucial and computationally intensive part of compression formats. Though the optimization technique is recognized, longer codes are generally not given priority

because they don't often appear. The original code update argued against this belief, and it was accepted.

The issue highlighted isn't something that just using a memory-safe language could prevent. It's a unique scenario where avoiding overflow checks is desired. However, while the actual solution didn't change the `ReadSymbol` function, ensuring the safety of the tight loop remains critical. Wrong justification for such safety measures can lead to problems.

Vendors Respond

Google was swift in its response, having already rolled out an update on its Stable and Extended stable channels. Specifically, versions 116.0.5845.187 for Mac and Linux and versions 116.0.5845.187/188 for Windows have received these crucial updates. These versions are set to be distributed gradually over the forthcoming days and weeks.

Mozilla is not far behind, with plans to release its update today for Firefox in version 117.0.1, ensuring its vast user base remains protected. Apple also [seems to have pushed an update](#) that indicates the update was intended for this very vulnerability.

Acknowledgments

The vulnerability was responsibly reported by the Apple Security Engineering and Architecture (SEAR) team in collaboration with The Citizen Lab at The University of Toronto's Munk School on September 6, 2023.

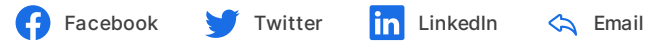
Moreover, it's worth noting that **Google has confirmed the existence of an exploit for CVE-2023-4863 in the wild**, emphasizing the urgency of the situation.

Users are urged to ensure their browsers are up-to-date with the latest versions to benefit from these crucial security patches.

About WebP

Developed by Google, WebP is a modern image format that offers superior lossless and lossy compression for images on the web. Given its advantages in size and speed over formats like PNG and JPEG, it has seen widespread adoption, making the discovery and rectification of this vulnerability all the more crucial.

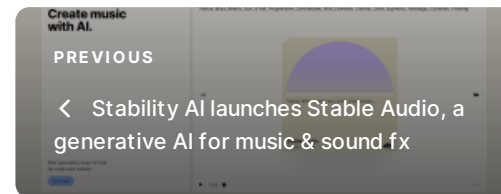
SHARE



WRITTEN BY

Alex Ivanovs

Alex is a full-stack developer with more than 15 years of experience. After many years of threading the self-taught path, he discovered a natural passion for writing. His past work includes helping build the Huffington Post Code column and working with publishers such as Entrepreneur, TheNextWeb, and many prominent tech startups.



READ ALSO

- [Stability AI launches Stable Audio, a generative AI for music & sound fx](#)
- [The 10 Best Tools for Automated Marketing & Behavioral Emails](#)
- [The 10 Best Open-Source Headless CMS](#)
- [10 Tips for Optimizing NGINX Performance](#)
- [Uncaught TypeError: Cannot read property of undefined](#)



© 2023 STACK DIARY - ALL RIGHTS RESERVED.

[ABOUT](#)

[ADVERTISE](#)

[DISCLOSURE](#)

[CONTACT](#)