Intigriti insights: React2Shell CVE-2025-55182

BY ELEANOR BARLOW · DECEMBER 5, 2025 · LAST UPDATED ON DECEMBER 24, 2025

What you will learn

- **How the React2Shell (CVE-2025-55182) vulnerability works** and why it enables remote code execution in React Server Components.
- **Which React applications are vulnerable,** and how to assess whether your stack is affected.
- **How to reduce risk and respond effectively**, including patching guidance and the role of security testing programs.

This blog explores the widespread and critical state of the React2Shell vulnerability. It provides a technical overview, suggested mitigations, and actions to safeguard people, processes, and data, as well as a review of what our team has experienced and seen off the back of this vulnerability.

Please note that as more is learnt, Intigriti continues to update reports, provide information on what our triage team and researchers are seeing, and will be regularly updating this content with the latest patches and fixes.

What is CVE-2025-55182, and when was it discovered?

On the 3rd of December 2025, React, which is an open-source front-end JavaScript library, released a blog entitled '**Critical Security Vulnerability in React Server Components'** which details a maximum severity vulnerability (CVSS 10).

This vulnerability (CVE-2025-55182) has been named React2Shell (the name being a gesture to Log4Shell).

According to the release, React2Shell was reported on the 29th of November. 'Lachlan Davidson reported a security vulnerability in React that allows unauthenticated remote code execution by exploiting a flaw in how React decodes payloads sent to React Server Function endpoints.'

On December 1st, a fix was created, and the React team began to implement mitigations and rolled out the fix. December 3rd, and the fix was published and publicly disclosed as CVE-2025-55182.

What has been impacted, and who is at risk?

At the time the blog was released, the vulnerability was present in versions 19.0, 19.1.0, 19.1.1, and 19.2.0 of react-server-dom-webpack, react-server-dom-parcel, and react-server-dom-turbopack.

An important thing to note here is that even if your app does not implement any React Server Function endpoints, the blog highlights that you may still be vulnerable if your app supports React Server

Components.

There are numerous mechanisms to identify if an asset is running React Server Components (RSC), but identifying presence is not enough to determine whether an asset is vulnerable.

How does it work?

React aims to make building user interfaces based on components seamless. React delivers integration points, as well as tools, that frameworks use to run code. What React does is translate the client's HTTP requests, which are then forwarded to the server. There, on the server, the HTTP request is translated into a function call to return the required data to the client.

Now, with React2Shell, a malicious HTTP request could be crafted and sent to a server function endpoint, and, when reserialized by React, can then execute remote code execution on said server.

Intigriti findings

What we, the Intigriti team, have noticed off the back of this is an increase in the number of submissions in the triage queue based on this vulnerability.

Since the vulnerability was first reported, Intigriti has received over a hundred of total reports regarding React2Shell.

- The majority of these reports have been confirmed to be vulnerable and exploitable.
- While there is a cooldown period for recently discovered vulnerabilities, we are seeing companies reward bonuses to show their gratitude for being made aware of this issue.

Intigriti recommendations and tips to maximize your BB and VDP programs

First, we recommend that you patch to the latest versions and apply the latest updates. This blog provides patches for each of the impacted elements.

While your team may be fixing and patching internally, this is a great time to leverage expert researchers to ensure no elements are missed.

Tips to maximize the value brought to you by your Bug Bounty and Vulnerability Disclosure Programs (VDPs) include:

- Let researchers know where they should focus their efforts by keeping your policy page up to date.
- Consider incentivizing researchers to investigate and, if you feel like you have already remediated the issue, set up an additional incentive to uncover blind spots.
- While most submissions will be made via bug bounty programs, for VDPs, it is still worth updating your community if you are accepting React2Shell reports, as these may be submitted through VDPs as well.

Companies that are quick to move and remediate will have the best results in lowering the impact of this vulnerability.

For more information or to speak with a member of our team, contact us here.

AUTHOR

Eleanor Barlow

Eleanor Barlow is a London-based Senior Cyber Security Technical Writer at Intigriti, with 9+ years' experience reporting on and writing for the cyber and tech sector. She specializes in data-driven content on cybersecurity and bug bounty intelligence, helping organizations benefit from the latest trends and insights.