

China-nexus cyber threat groups rapidly exploit React2Shell vulnerability (CVE-2025-55182) | Amazon Web Services

Published: 2025-12-05 · Archived: 2026-07-28 02:00:16 UTC

December 29, 2025: The blog post was updated to add options for AWS Network Firewall.

December 12, 2025: The blog post was updated to clarify when customers need to update their ReactJS version.

Within hours of the public disclosure of CVE-2025-55182 (React2Shell) on December 3, 2025, Amazon threat intelligence teams observed active exploitation attempts by multiple China state-nexus threat groups, including Earth Lamia and Jackpot Panda. This critical vulnerability in React Server Components has a maximum Common Vulnerability Scoring System (CVSS) score of 10.0 and affects React versions 19.x and Next.js versions 15.x and 16.x when using App Router. While this vulnerability doesn't affect AWS services, we are sharing this threat intelligence to help customers running React or Next.js applications in their own environments take immediate action.

China continues to be the most prolific source of state-sponsored cyber threat activity, with threat actors routinely operationalizing public exploits within hours or days of disclosure. Through monitoring in our [AWS MadPot](#) honeypot infrastructure, Amazon threat intelligence teams have identified both known groups and previously untracked threat clusters attempting to exploit CVE-2025-55182. AWS has deployed multiple layers of automated protection through Sonaris active defense, [AWS WAF](#) managed rules ([AWSManagedRulesKnownBadInputsRuleSet version 1.24](#) or higher), and perimeter security controls. However, these protections aren't substitutes for patching. Regardless of whether customers are using a fully managed AWS service, if customers are running an affected version of React or Next.js in their environments, they should update to the latest patched versions immediately. Customers running React or Next.js in their own environments ([Amazon Elastic Compute Cloud \(Amazon EC2\)](#), containers, and so on) must update vulnerable applications immediately.

Understanding CVE-2025-55182 (React2Shell)

Discovered by Lachlan Davidson and disclosed to the React Team on November 29, 2025, CVE-2025-55182 is an unsafe deserialization vulnerability in React Server Components. The vulnerability was named *React2Shell* by security researchers.

Key facts:

- **CVSS score:** 10.0 (Maximum severity)
- **Attack vector:** Unauthenticated remote code execution
- **Affected components:** React Server components in React 19.x and Next.js 15.x/16.x with App Router

- **Critical detail:** Applications are vulnerable even if they don't explicitly use server functions, as long as they support React Server Components

The vulnerability was responsibly disclosed by Vercel to Meta and major cloud providers, including AWS, enabling coordinated patching and protection deployment prior to the public disclosure of the vulnerability.

Who is exploiting CVE-2025-55182?

Our analysis of exploitation attempts in AWS MadPot honeypot infrastructure has identified exploitation activity from IP addresses and infrastructure historically linked to known China state-nexus threat actors. Because of shared anonymization infrastructure among Chinese threat groups, definitive attribution is challenging:

- **Infrastructure associated with Earth Lamia:** Earth Lamia is a China-nexus cyber threat actor known for exploiting web application vulnerabilities to target organizations across Latin America, the Middle East, and Southeast Asia. The group has historically targeted sectors across financial services, logistics, retail, IT companies, universities, and government organizations.
- **Infrastructure associated with Jackpot Panda:** Jackpot Panda is a China-nexus cyber threat actor primarily targeting entities in East and Southeast Asia. The activity likely aligns to collection priorities pertaining to domestic security and corruption concerns.
- **Shared anonymization infrastructure:** Large-scale anonymization networks have become a defining characteristic of Chinese cyber operations, enabling reconnaissance, exploitation, and command-and-control activities while obscuring attribution. These networks are used by multiple threat groups simultaneously, making it difficult to attribute specific activities to individual actors.

This is in addition to many other unattributed threat groups that share commonality with Chinese-nexus cyber threat activity. The majority of observed autonomous system numbers (ASNs) for unattributed activity are associated with Chinese infrastructure, further confirming that most exploitation activity originates from that region. The speed at which these groups operationalized public proof-of-concept (PoC) exploits underscores a critical reality: when PoCs hit the internet, sophisticated threat actors are quick to weaponize them.

Exploitation tools and techniques

Threat actors are using both automated scanning tools and individual PoC exploits. Some observed automated tools have capabilities to deter detection such as user agent randomization. These groups aren't limiting their activities to CVE-2025-55182. Amazon threat intelligence teams observed them simultaneously exploiting other recent N-day vulnerabilities, including CVE-2025-1338. This demonstrates a systematic approach: threat actors monitor for new vulnerability disclosures, rapidly integrate public exploits into their scanning infrastructure, and conduct broad campaigns across multiple Common Vulnerabilities and Exposures (CVEs) simultaneously to maximize their chances of finding vulnerable targets.

The reality of public PoCs: Quantity over quality

A notable observation from our investigation is that many threat actors are attempting to use public PoCs that don't actually work in real-world scenarios. The GitHub security community has identified multiple PoCs that

demonstrate fundamental misunderstandings of the vulnerability:

- Some of the example exploitable applications explicitly register dangerous modules (`fs` , `child_process` , `vm`) in the server manifest, which is something real applications should never do.
- Several repositories contain code that would remain vulnerable even after patching to safe versions.

Despite the technical inadequacy of many public PoCs, threat actors are still attempting to use them. This demonstrates several important patterns:

- **Speed over accuracy:** Threat actors prioritize rapid operationalization over thorough testing, attempting to exploit targets with any available tool.
- **Volume-based approach:** By scanning broadly with multiple PoCs (even non-functional ones), actors hope to find the small percentage of vulnerable configurations.
- **Low barrier to entry:** The availability of public exploits, even flawed ones, enables less sophisticated actors to participate in exploitation campaigns.
- **Noise generation:** Failed exploitation attempts create significant noise in logs, potentially masking more sophisticated attacks.

Persistent and methodical attack patterns

Analysis of data from MadPot reveals the persistent nature of these exploitation attempts. In one notable example, an unattributed threat cluster associated with IP address 183[.]6.80.214 spent nearly an hour (from 2:30:17 AM to 3:22:48 AM UTC on December 4, 2025) systematically troubleshooting exploitation attempts:

- 116 total requests across 52 minutes
- Attempted multiple exploit payloads
- Tried executing Linux commands (`whoami` , `id`)
- Attempted file writes to `/tmp/pwned.txt`
- Tried to read `/etc/passwd`

This behavior demonstrates that threat actors aren't just running automated scans, but are actively debugging and refining their exploitation techniques against live targets.

How AWS helps protect customers

AWS deployed multiple layers of protection to help safeguard customers:

- **Sonaris Active Defense**

Our Sonaris threat intelligence system automatically detected and restricted malicious scanning attempts targeting this vulnerability. Sonaris analyzes over 200 billion events per minute and integrates threat intelligence from our MadPot honeypot network to identify and block exploitation attempts in real time.

- **MadPot Intelligence**

Our global honeypot system provided early detection of exploitation attempts, enabling rapid response and threat analysis.

- **AWS WAF Managed Rules**

The default version (1.24 or higher) of the AWS WAF `AWSManagedRulesKnownBadInputsRuleSet` now includes updated rules for CVE-2025-55182, providing automatic protection for customers using AWS WAF with managed rule sets.

- **AWS Network Firewall Rule Options**

Managed

The [Active Threat Defense managed rules](#) for AWS Network Firewall are automatically updated with the latest threat intelligence from MadPot so customers can get proactive protection for their VPCs.

Custom

The following AWS Network Firewall custom L7 stateful rule blocks HTTP connections made directly to IP addresses on non-standard ports (any port other than 80). This pattern has been commonly observed by Amazon Threat Intelligence in post-exploitation scenarios where malware downloads additional payloads or establishes command-and-control communications by connecting directly to IP addresses rather than domain names, often on high-numbered ports to evade detection.

While not necessarily specific to React2Shell, many React2Shell exploits include this behavior, which is usually anomalous in most production environments. You can choose to block and log these requests or simply alert on them so you can investigate systems that are triggering the rule to determine whether they have been affected.

```
reject http $HOME_NET any -> any !80 (http.host; content:"."; pcre:"/^(?:[0-9]{1,3}\.){3}[0-9]{1,3}$/"; msg:"Direct to IP HTTP on non-standard port (common post exploitation malware download technique)"; flow:to_server; sid:2025121801;)
```

- **Amazon Threat Intelligence**

Amazon threat intelligence teams are actively investigating CVE-2025-55182 exploitation attempts to protect AWS infrastructure. If we identify signs that your infrastructure has been compromised, we will notify you through AWS Support. However, application-layer vulnerabilities are difficult to detect comprehensively from network telemetry alone. Do not wait for notification from AWS.

Important: These protections are not substitutes for patching. Customers running React or Next.js in their own environments (EC2, containers, etc.) must update vulnerable applications immediately.

Immediate recommended actions

1. Update vulnerable React/Next.js applications. See the AWS Security Bulletin (<https://aws.amazon.com/security/security-bulletins/AWS-2025-030/>) for affected and patched versions.
2. Deploy the custom AWS WAF rule as interim protection (rule provided in the security bulletin).
3. Review application and web server logs for suspicious activity.
4. Look for POST requests with `next-action` or `rsc-action-id` headers.
5. Check for unexpected process execution or file modifications on application servers.

If you believe your application may have been compromised, [open an AWS Support case immediately for assistance with incident response](#).

Note: Customers using managed AWS services are not affected and require no action.

Indicators of compromise

Network indicators

- HTTP POST requests to application endpoints with `next-action` or `rsc-action-id` headers
- Request bodies containing `$@` patterns
- Request bodies containing `"status":"resolved_model"` patterns

Host-based indicators

- Unexpected execution of reconnaissance commands (`whoami` , `id` , `uname`)
- Attempts to read `/etc/passwd`
- Suspicious file writes to `/tmp/` directory (for example, `pwned.txt`)
- New processes spawned by Node.js/React application processes

Threat actor infrastructure

IP Address, Date of Activity, Attribution

206[.]237.3.150, 2025-12-04, Earth Lamia

45[.]77.33.136, 2025-12-04, Jackpot Panda

143[.]198.92.82, 2025-12-04, Anonymization Network

183[.]6.80.214, 2025-12-04, Unattributed threat cluster

Additional resources

- AWS Security Bulletin: CVE-2025-55182 <https://aws.amazon.com/security/security-bulletins/AWS-2025-030/>
- AWS WAF Documentation: <https://docs.aws.amazon.com/waf/>
- React Team Security Advisory: <https://react.dev/blog/2025/12/03/react-server-components-security-advisory>

If you have feedback about this post, submit comments in the **Comments** section below. If you have questions about this post, [contact AWS Support](#).

Source: <https://aws.amazon.com/blogs/security/china-nexus-cyber-threat-groups-rapidly-exploit-react2shell-vulnerability-cve-2025-55182/>