TLP: CLEAR

RELATÓRIO DE PESQUISAS

React2Shell: Mecanismos de ataque e estratégias de defesa

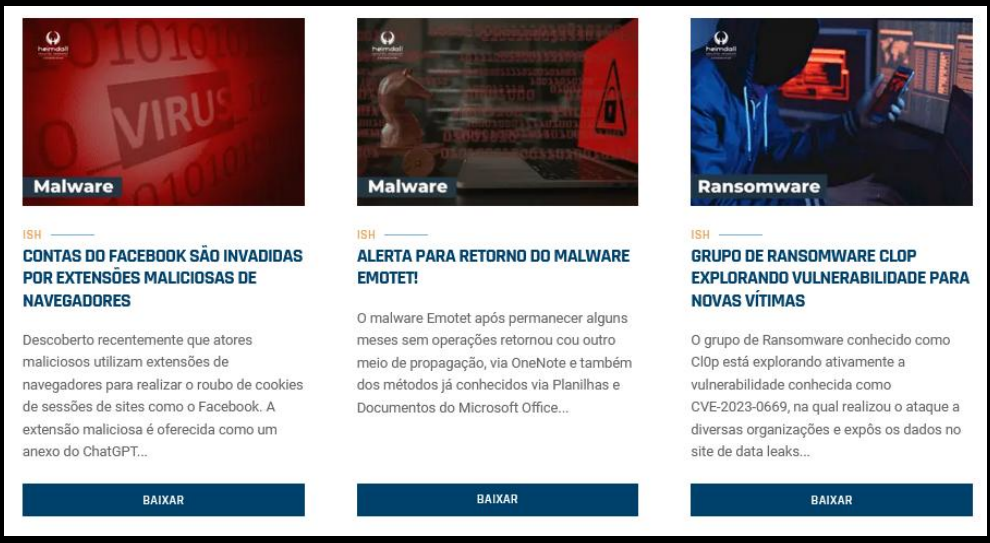
heimdall
security research

Acesse a nossa nova comunidade através do WhatsApp!

Heimdall Security Research

Acesse boletins diários sobre agentes de ameaças, *malwares*, indicadores de comprometimentos, TTPs e outras informações no *site* da ISH.

Boletins de Segurança – Heimdall



SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO EXECUTIVA

React2Shell é o nome dado à exploração da vulnerabilidade crítica catalogada na CVE-2025-55182, que permite execução remota de código sem autenticação em aplicações que usam React Server Components (RSC). O problema está ligado ao processamento de payloads do protocolo Flight e pode ser explorado com uma única requisição HTTP, executando comandos com o mesmo privilégio do processo do servidor web, geralmente Node.js. O risco é amplificado porque ecossistemas populares, como aplicações Next.js com App Router e RSC habilitado, podem estar expostos em configurações comuns de produção. A exploração já foi observada "na vida real" pouco após a divulgação pública, com uso por múltiplos grupos e cadeias de malware.

2 ESTRATÉGICO

2.1 INTRODUÇÃO

Do ponto de vista estratégico, React2Shell é um risco de comprometimento total do servidor. Como a falha permite RCE pré autenticação, ela acelera o ciclo de ataque: descoberta de alvo exposto, exploração, implantação de payload, persistência e movimentação lateral. Isso reduz a dependência de credenciais e burla controles tradicionais que focam somente em login, MFA e hardening de contas.

2.2 VITIMOLOGIA E SEGMENTOS IMPACTADOS

O impacto é mais significativo em organizações que mantêm aplicações públicas utilizando React Server Components, especialmente em ambientes com ampla superfície de exposição na internet e processos de atualização pouco ágeis. Evidências de exploração indicam campanhas ativas atingindo diferentes setores e regiões, combinando estratégias de monetização rápida, como a implantação de cryptominers, com operações mais avançadas focadas em espionagem e persistência de longo prazo.

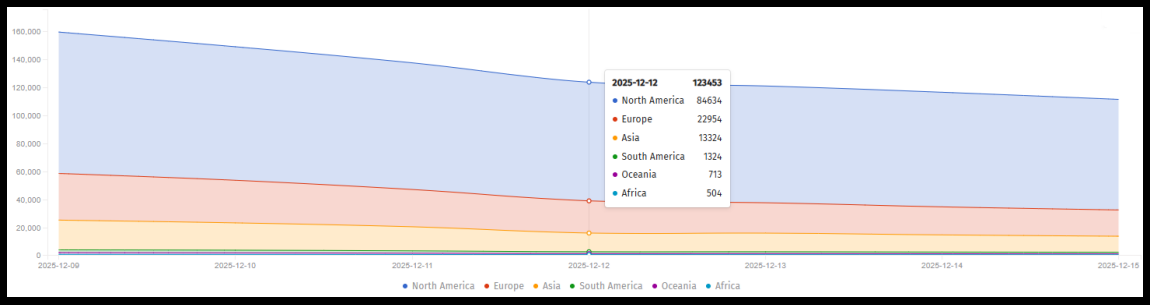
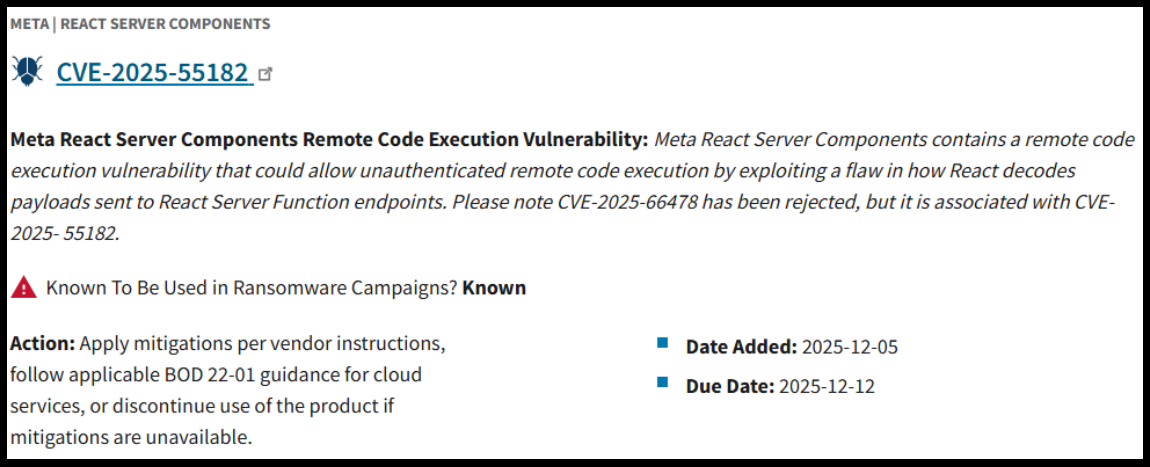


Figura 1 – Distribuição de impacto da falha por continente.


META | REACT SERVER COMPONENTS

CVE-2025-55182

Meta React Server Components Remote Code Execution Vulnerability: *Meta React Server Components contains a remote code execution vulnerability that could allow unauthenticated remote code execution by exploiting a flaw in how React decodes payloads sent to React Server Function endpoints. Please note CVE-2025-66478 has been rejected, but it is associated with CVE-2025- 55182.*

Known To Be Used in Ransomware Campaigns? **Known**

Action: Apply mitigations per vendor instructions, follow applicable BOD 22-01 guidance for cloud services, or discontinue use of the product if mitigations are unavailable.

- **Date Added:** 2025-12-05
- **Due Date:** 2025-12-12

Figura 2 – Vulnerabilidade adicionada ao catalogo KEV-CISA.

3 TÁTICO

3.1 O QUE É A VULNERABILIDADE E POR QUE ELA ACONTECE

A CVE-2025-55182 é descrita como uma vulnerbailidade crítica em como o servidor decodifica e desserializa payloads enviados para endpoints ligados a React Server Components e Server Functions. Um adversário envia um payload especialmente construído que leva o runtime a interpretar dados não confiáveis de forma perigosa, resultando em execução de código no servidor.

Em termos práticos, o atacante não precisa de sessão nem de conta. O alvo típico é uma aplicação exposta que processa requisições RSC, muitas vezes em rotas e endpoints internos do framework, e que executa com permissões suficientes para criar arquivos, abrir conexões de rede e disparar processos filhos.

3.2 COMO A EXPLORAÇÃO FUNCIONA NA PRÁTICA

O fluxo tático mais comum segue esta lógica:

- O atacante identifica uma aplicação exposta com indícios de RSC e versões vulneráveis do React 19.x usadas no servidor
- Ele envia uma requisição HTTP com payload do protocolo Flight malformado ou malicioso para um endpoint processado por RSC
- A aplicação desserializa o conteúdo e o atacante obtêm execução remota de código no contexto do processo do servidor
- Após o RCE, o atacante executa comandos para reconhecimento, exfiltração e/ou persistência.

3.3 TABELA MITRE ATT&CK

Este tópico apresenta as Táticas, Técnicas e Procedimentos (TTPs) identificados nesta ameaça, conforme o framework MITRE ATT&CK, oferecendo uma visão tática detalhada sobre o comportamento do adversário. O objetivo é permitir o mapeamento das técnicas utilizadas pelos atacantes, facilitando a implementação de contramedidas eficazes e o aprimoramento das defesas de segurança.

Tática	Técnica	Detalhes
Initial Access	T1190 Exploit Public Facing Application	Exploração de aplicação web exposta para obter execução inicial.

Tabela 1 – Tabela MITRE ATT&CK.

4 OPERACIONAL

4.1 EMULAÇÃO CONTROLADA

Para validar a exposição e preparar mecanismos de detecção, a emulação deve ser conduzida em ambiente isolado e devidamente autorizado. O objetivo operacional não é simplesmente comprovar a execução remota de código, mas verificar se a aplicação aceita e processa payloads associados à vulnerabilidade, bem como identificar quais logs e sinais de telemetria são gerados durante tentativas de exploração. Estudos públicos indicam que a falha pode ser explorada mesmo em configurações padrão em determinados cenários, o que reforça a necessidade de testes de segurança contínuos antes da exposição da aplicação em ambiente de produção.

4.2 COMO GRUPOS APT E CAMPANHAS ESTÃO USANDO REACT2SHELL

Observações recentes indicam exploração por múltiplos atores, incluindo operações ligadas a estados e grupos oportunistas. Um padrão recorrente é o uso inicial para entrega de mineradores e, em casos mais sofisticados, para backdoors e implantes persistentes. Há campanhas atribuídas a atores ligados à Coreia do Norte que exploraram React2Shell para implantar um backdoor com comando e controle apoiado em artefatos do ecossistema Ethereum, buscando persistência e acesso contínuo.

Também há relatos de exploração por atores associados à China e por atividade ligada ao Irã, além de discussões e compartilhamento de ferramentas de varredura e PoCs em fóruns clandestinos, acelerando o volume de tentativas de exploração em massa.

4.3 Métodos de detecção

A detecção eficaz exige a correlação de telemetria proveniente da aplicação, do servidor e da camada de rede, uma vez que a tentativa de exploração pode se apresentar inicialmente como uma requisição web legítima, só revelando comportamento malicioso após a execução do payload no servidor.

- **Detecção na camada HTTP e de aplicação**: É recomendável correlacionar picos de requisições anômalas direcionadas a endpoints associados a React Server Components e Server Functions com respostas HTTP 500, timeouts ou aumentos súbitos de latência. Também devem ser analisados padrões incomuns de payload e tamanhos atípicos em requisições POST que acionam o processamento do protocolo Flight. Esses sinais devem ser integrados ao SIEM por meio de logs de proxy reverso, WAF e da própria aplicação, permitindo correlação por endereço IP, user agent, frequência de requisições e códigos de resposta.
- **Detecção em host e runtime Node.js**: No nível de host, alertas devem ser configurados para identificar a criação de processos filhos a partir do processo do Node.js, especialmente shells e ferramentas de sistema como sh, bash, curl, wget, python, perl e netcat. Indicadores adicionais incluem modificações suspeitas em diretórios de build e runtime, criação de mecanismos de persistência como cron ou systemd, descarte de binários temporários e execução de scripts de bootstrap. Conexões de saída inéditas ou direcionadas a destinos raros logo após uma sequência de erros HTTP na aplicação também devem ser tratadas como sinais de possível comprometimento.
- **Detecção em ambientes conteinerizados e cloud**: Em ambientes baseados em containers, é fundamental aplicar políticas de runtime que impeçam a execução de shells e ferramentas de download a partir do processo da aplicação, gerando alertas sempre que essas tentativas ocorrerem. O monitoramento contínuo de escrita em caminhos sensíveis e da criação de executáveis em volumes persistentes é essencial para identificar tentativas de persistência e movimentos pós exploração.

4.4 Correções

A correção prioritária consiste na atualização imediata das versões afetadas. Recomendações oficiais indicam a necessidade de atualizar o React para versões corrigidas, como 19.0.1, 19.1.2 ou 19.2.1, conforme a linha utilizada, além de manter o framework que implementa React Server Components, como o Next.js, sempre atualizado com os últimos patches de segurança.

Também é importante considerar vulnerabilidades correlatas divulgadas após a correção inicial, incluindo falhas de negação de serviço e ajustes pós patch,

que exigem atualizações adicionais para reduzir risco residual e impacto operacional.

Quando a atualização imediata não for possível, devem ser adotadas mitigações compensatórias, como restringir a exposição de rotas e endpoints de React Server Components por meio de controles de borda, reforçar regras de WAF para bloquear padrões de payload maliciosos, executar o serviço com o menor privilégio possível e habilitar monitoramento de integridade, limitando a execução de ferramentas administrativas no ambiente de produção.

5 Conclusão

React2Shell representa um cenário crítico para aplicações modernas que utilizam React Server Components, ao permitir execução remota de código sem autenticação e viabilizar exploração rápida e em larga escala. A vulnerabilidade já foi observada tanto em campanhas oportunistas quanto em operações conduzidas por grupos avançados. Nesse contexto, a velocidade de aplicação de correções é o fator mais relevante para a redução do risco, uma vez que o intervalo entre a divulgação pública e a exploração ativa foi curto e a disseminação de PoCs e ferramentas automatizadas tende a aumentar a pressão sobre ambientes não atualizados.

6 RECOMENDAÇÕES

Governança e atualização

- É fundamental inventariar todas as aplicações que utilizam React 19.x no lado servidor e identificar onde React Server Components estão habilitados, priorizando aquelas expostas à internet. As versões corrigidas do React e as atualizações de segurança do framework devem ser aplicadas com urgência, após validação em ambiente de homologação, e promovidas rapidamente para produção.

Hardening e redução de impacto

- A aplicação deve ser executada seguindo o princípio de menor privilégio, com controle rigoroso de egress e sem armazenamento de segredos no filesystem do container. Controles de runtime devem ser implementados para impedir a criação de shells e o uso de ferramentas de download a partir do processo da aplicação, gerando alertas sempre que houver tentativas desse tipo.

Detecção e resposta

- Devem ser criadas regras no SIEM que correlacionem erros de aplicação com a criação de processos filhos e o surgimento de novas conexões de saída. Além disso, é essencial manter playbooks de resposta atualizados, contemplando contenção rápida, coleta de evidências, rotação de segredos, verificação de persistência e validação da integridade dos sistemas afetados, certificados anômalos e comportamentos de autenticação suspeitos.

6.1 Indicadores de Comprometimento (IoC)

Indicadores do artefato	
md5:	8a8951ffcbede6f4bedff6f3191179fd
sha1:	8bb6514ac3935479902820d0486df8b6abee73dd
sha256:	df3f20a961d29eed46636783b71589c183675510737c984a11f78932b177b540
File name:	ma0oqd17.exe

Indicadores do artefato	
md5:	10231e4c2ade4f21c9d4fa52cabc8b5e
sha1:	972fe0233cea777f69ba5f081d60219eba73c617
sha256:	92064e210b23cf5b94585d3722bf53373d54fb4114dca25c34e010d0c010edf3
File name:	bf972l.exe

Indicadores do artefato	
md5:	1b0de91ffdb3bbb71c9cbf37c31299d0
sha1:	b66e7b8f153779ae8521248b502fcf5e5116b3af
sha256:	a455731133c00fdd2a141bdfba4def34ae58195126f762cdf951056b0ef161d4
File name:	a455731133c00fdd2a141bdfba4def34ae58195126f762cdf951056b0ef161d4

Tabela 2 – Indicadores de Comprometimento.

Indicadores de URL, IPs e Domínios

Indicadores de URL, IPs e Domínios	
URL	hxxp[:]//46[.]36[.]37[.]85:12000/sex[.]sh
Domínio	reactcdn[.]windowserrorapis[.]com
IP	193.142[.]147[.]209 82.163[.]22[.]139 216.158[.]232[.]43 45.76[.]155[.]14 115[.]42[.]60[.]223 54.39[.]28[.]147

Tabela 3 – Indicadores de Comprometimento.

Obs: Os *links* e endereços IP elencados acima podem estar ativos; cuidado ao realizar a manipulação dos referidos IoCs, evite realizar o clique e se tornar vítima do conteúdo malicioso hospedado no IoC.

7 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- MITRE ATT&CK
- CVE
- KEV-CISA
- Shadowserver

8 AUTORES

- **Cleriston de Freitas Santos Portela – Threat Researcher**

heimdall
security research
A DIVISION OF ISH