Escuela Politécnica Nacional

Facultad de Ingeniería de Sistemas

Aplicaciones Web

Nombre: Joel Patricio Quilumba Morocho
Paralelo: GR2CC
Fecha: 6 de enero de 2026

1. Identificación del párrafo con la explicación técnica

Después de leer detenidamente todos los textos, he identificado que el **Párrafo 5** es el que contiene la explicación técnica precisa sobre cómo se logra la ejecución remota de código.

Párrafo 5
Dentro de los detalles técnicos, se menciona que React Server Components utiliza un mecanismo de serialización y deserialización para transmitir estados entre cliente y servidor. Cuando este proceso no valida adecuadamente los datos recibidos, un atacante puede inyectar estructuras manipuladas que, al ser interpretadas por el servidor, terminan ejecutando instrucciones arbitrarias. Aunque en los comunicados oficiales se habla de "fallo de seguridad en la comunicación", lo que realmente ocurre es que la deserialización insegura abre la puerta a la ejecución remota de código, incluso sin credenciales previas.

2. Explicación de la causa técnica

Basándome en lo leído, la vulnerabilidad se produce por un fallo conocido como **deserialización insegura**.

Lo entiendo de esta forma: para que React Server Components funcione, necesita enviar información compleja del servidor al cliente y viceversa. Para hacer esto posible, el sistema convierte los objetos de datos en un formato transmisible (proceso de serialización) y luego los reconstruye al recibirlos (proceso de deserialización).

El problema técnico es que el servidor **confía ciegamente** en los datos que recibe para reconstruirlos, sin validar previamente si son seguros o legítimos. Esta falta de verificación permite que un atacante envíe datos manipulados o "tramposos" que contienen instrucciones ocultas. Cuando el servidor intenta procesar (deserializar) estos datos, termina ejecutando esas instrucciones maliciosas automáticamente, lo que le da al atacante control sobre el servidor sin necesitar siquiera una contraseña.