AIRNITE

ZERO DAY

NÚMERO 04/2026 · SEGURIDAD INFORMÁTICA

Shai-Hulud: el gusano que devoró las bibliotecas de JavaScript

En 2025, un gusano que se replicaba solo infectó cientos de paquetes npm, robó credenciales de desarrolladores y las publicó en GitHub. Cómo funcionó el ataque y qué debe hacer ahora cualquier equipo de desarrollo.

Publicado: 21 de noviembre de 2026 Tiempo de lectura: 6 min Autor: Redacción de Airnite

SOFTWARE. SECURITY. CODE.

// LA SITUACIÓN EN CIFRAS

500+
paquetes comprometidos (1.ª oleada)
CISA, 23/09/2025; GitHub retiró más de 500 paquetes

≈ 500
paquetes en la 2.ª oleada
NCSC de Nueva Zelanda, 25/11/2025

16/09/2025
fecha de corte de versiones seguras
CISA: fijar versiones anteriores a esa fecha

7 días
vida útil de los nuevos tokens de npm
medida prevista por GitHub (22/09/2025)

npm es el registro central de paquetes de JavaScript. Muchas aplicaciones web modernas toman de él numerosas piezas de terceros. En septiembre de 2025, GitHub, que gestiona npm, informó de un ataque que se propagaba como un gusano de paquete en paquete. Su nombre procede de la novela «Dune»: **Shai-Hulud**, el gusano de arena.

// ¿Qué ha pasado?

Según GitHub, la empresa tuvo noticia del ataque el 14 de septiembre de 2025. El gusano, que se replicaba solo, había introducido scripts maliciosos en paquetes populares. GitHub afirma que retiró del registro **más de 500 paquetes comprometidos** y bloqueó las subidas con los indicadores conocidos del malware. El 23 de septiembre de 2025, la agencia estadounidense CISA también advirtió de que el gusano había comprometido más de 500 paquetes.

En noviembre llegó una segunda oleada. El 25 de noviembre de 2025, el NCSC de Nueva Zelanda alertó de una versión evolucionada que afectaba a casi 500 paquetes más, con una novedad peligrosa: si el código malicioso no encontraba credenciales aprovechables, según el NCSC **borraba todos los archivos del directorio personal** del usuario.

// ¿Cómo funciona el ataque?

El gusano se aprovecha de la confianza en las actualizaciones de paquetes. Cuando un desarrollador o un sistema de compilación instala un paquete infectado, se ejecuta un script durante la instalación. Según la CISA, este busca credenciales en el entorno, en particular **tokens de acceso de GitHub** y claves de las nubes de AWS, Google y Microsoft Azure. Envía el botín a los atacantes y además lo sube a un repositorio público de GitHub llamado «Shai-Hulud».

Luego llega la parte de gusano: con los accesos robados, el código malicioso inicia sesión en npm como el desarrollador afectado, según la CISA, se introduce en sus otros paquetes y publica nuevas versiones infectadas. Quien las instala se convierte en la siguiente víctima. En la segunda oleada, según el NCSC, el código instalaba además el entorno de ejecución de JavaScript Bun para ejecutarse y guardaba los datos robados en repositorios con nombres aleatorios.

// ¿A quién afecta?

Afecta a toda empresa que desarrolle o compile software en JavaScript, es decir, a casi cualquier empresa con aplicación web, tienda online o app propia. Los servidores de compilación y los equipos de desarrollo con credenciales de la nube y de producción corren un riesgo especial. Instalar allí un paquete infectado puede, en el peor de los casos, entregar las llaves de toda la infraestructura.

También las empresas que solo compran software se ven afectadas indirectamente: sus proveedores trabajan con los mismos paquetes. Por eso, la pregunta para cualquier socio de software es: ¿cómo protegéis vuestro entorno de compilación y vuestras credenciales?

// ¿Qué hay que hacer ahora?

La CISA recomienda revisar todos los proyectos con dependencias npm, fijar las dependencias en versiones seguras publicadas **antes del 16 de septiembre de 2025**, renovar de inmediato todas las credenciales de desarrollo y exigir **autenticación multifactor resistente al phishing** en todas las cuentas de desarrollo. El NCSC aconseja además desactivar, cuando sea posible, los scripts de instalación en entornos CI y congelar temporalmente las actualizaciones de paquetes.

GitHub anunció que reforzaría a fondo la publicación de paquetes: solo con autenticación de dos factores, con tokens válidos como máximo **siete días** o mediante «trusted publishing» directamente desde sistemas de compilación verificados. La lista de control de abajo resume los pasos para su equipo.

// ¿Cómo ayuda Airnite?

Airnite desarrolla **software** en un entorno de compilación protegido: versiones de paquetes fijadas, claves separadas para compilación y producción, revisión automática de nuevas dependencias. En una **prueba de penetración** comprobamos qué secretos encontraría un paquete comprometido en su entorno. En la **formación**, los equipos de desarrollo aprenden a reconocer los riesgos de la cadena de suministro y a manejar los tokens con seguridad.

> Nota: estado de la investigación a 21 de noviembre de 2026. No hemos comprobado si GitHub ha aplicado ya por completo todas las medidas anunciadas.

Lista de control: qué hacer ahora

[+] Revise todos los proyectos con dependencias npm, incluidas las anidadas en package-lock.json o yarn.lock.

[+] Fije las dependencias en versiones conocidas y revisadas y apruebe las actualizaciones de forma consciente en lugar de aceptarlas automáticamente.

[+] Ante cualquier sospecha, renueve de inmediato todas las credenciales de desarrollo, GitHub, npm, nube y CI/CD.

[+] Exija autenticación multifactor resistente al phishing (p. ej. llaves FIDO) en todas las cuentas de desarrollo.

[+] Desactive los scripts de instalación de paquetes en los sistemas CI cuando sea posible y ejecute las compilaciones sin acceso a claves de producción.

[+] Active en GitHub el escaneo de secretos, la protección de ramas y las actualizaciones de seguridad de Dependabot; elimine apps y webhooks innecesarios.

Cómo ayuda Airnite

- Software y sitios web – Sitios web y aplicaciones web: rápidos, seguros, multilingües.
- Pruebas de penetración y seguridad – Ataques controlados con un informe claro y un plan de medidas.
- Formación – Su equipo aprende nuestras herramientas y procesos.

¿Están protegidos los accesos de sus desarrolladores?

Airnite revisa su entorno de compilación y desarrollo: qué tokens y claves de la nube hay en los equipos y sistemas CI, cómo se incorporan los paquetes y si un paquete comprometido podría llegar a sus secretos. Recibe medidas concretas, desde la rotación de tokens hasta versiones de paquetes fijadas.

Solicitar una primera conversación · Escribir un correo · Por WhatsApp

Contacto

¿Preguntas sobre este artículo? Escríbanos por WhatsApp: +49 174 6834842

Leer en línea: https://airnite.de/es/revista/shai-hulud-gusano-npm/

Fuentes

Cada dato de este artículo se basa en las fuentes indicadas a continuación. Las fuentes primarias están señaladas.

1. **CISA · Widespread Supply Chain Compromise Impacting npm Ecosystem** · 23/09/2025 · FUENTE PRIMARIA
   Alerta de las autoridades de EE. UU.: más de 500 paquetes, comportamiento del gusano, recomendaciones
   https://www.cisa.gov/news-events/alerts/2025/09/23/widespread-supply-chain-compromise-impacting-npm-ecosystem
2. **NCSC de Nueva Zelanda · New wave of widespread supply chain compromise impacting npm ecosystem** · 25/11/2025 · FUENTE PRIMARIA
   Alerta de las autoridades: segunda oleada con casi 500 paquetes, entorno Bun, borrado del directorio personal
   https://www.ncsc.govt.nz/alerts/supply-chain-compromise-impacting-npm-ecosystem/
3. **GitHub Blog · Our plan for a more secure npm supply chain** · 22/09/2025 · FUENTE PRIMARIA
   Operador de la plataforma: retirada de más de 500 paquetes, medidas previstas (2FA, tokens de corta duración, trusted publishing)
   https://github.blog/security/supply-chain-security/our-plan-for-a-more-secure-npm-supply-chain/

Información editorial general, no constituye asesoramiento jurídico ni una evaluación de sus sistemas concretos. Vigencia: fecha de publicación.

Airnite GmbH · Am Waldbach 7 · 77654 Offenburg · airnite.de