Investigadores de ciberseguridad están lanzando una alerta sobre una nueva campaña de ataque a la cadena de suministro que apunta a paquetes npm vinculados con SAP, incorporando malware diseñado para robar credenciales.

De acuerdo con informes de Aikido Security, SafeDep, Socket, StepSecurity y Wiz (propiedad de Google), la campaña —autodenominada mini Shai-Hulud— ha impactado los siguientes paquetes relacionados con el ecosistema de JavaScript y desarrollo de aplicaciones en la nube de SAP:

- mbt@1.2.48
- @cap-js/db-service@2.10.1
- @cap-js/postgres@2.2.2
- @cap-js/sqlite@2.2.2

«Las versiones afectadas incorporaron comportamientos en el momento de la instalación que antes no formaban parte de la funcionalidad esperada de estos paquetes,» señaló Socket. *«Las ediciones comprometidas añadieron un script de preinstalación que funciona como un inicializador en tiempo de ejecución, descargando desde GitHub Releases un archivo ZIP de Bun específico para cada plataforma, extrayéndolo y ejecutando de inmediato el binario obtenido.»*

«Además, la implementación sigue redirecciones HTTP sin comprobar el destino y emplea PowerShell con -ExecutionPolicy Bypass en Windows, lo que incrementa el riesgo en entornos de desarrollo y CI/CD afectados.»

Wiz indicó que los paquetes maliciosos presentan varias características observadas en operaciones previas de TeamPCP, lo que sugiere que probablemente el mismo actor de amenazas está detrás de esta campaña reciente.

Las versiones sospechosas fueron publicadas el 29 de abril de 2026, entre las 09:55 UTC y las 12:14 UTC. Los paquetes comprometidos introducen un nuevo hook de preinstalación en package.json que ejecuta un archivo llamado «setup.mjs», el cual actúa como cargador del

entorno de ejecución JavaScript Bun para activar el ladrón de credenciales y el framework de propagación («execution.js»).

Según Aikido, el malware está diseñado para recolectar credenciales locales de desarrolladores, tokens de GitHub y npm, secretos de GitHub Actions y credenciales en la nube provenientes de AWS, Azure, GCP y Kubernetes. La información robada se cifra y se extrae hacia repositorios públicos de GitHub creados dentro de la propia cuenta de la víctima, con la descripción *«A Mini Shai-Hulud has Appeared.»* Hasta el momento, existen más de 1,100 repositorios con dicha descripción.

Adicionalmente, la carga útil de 11.6 MB incluye funciones de autopropagación a través de flujos de trabajo de desarrollo y publicación. En concreto, utiliza tokens de GitHub y npm para insertar un flujo de trabajo malicioso de GitHub Actions en los repositorios de la víctima, con el fin de robar secretos y publicar versiones comprometidas de paquetes npm en el registro.

No obstante, este incidente presenta diferencias relevantes frente a oleadas anteriores de Shai-Hulud:

- Todos los datos extraídos se cifran con AES-256-GCM y la clave se encapsula mediante RSA-4096 con una clave pública integrada en la carga útil, lo que hace que solo el atacante pueda descifrarlos.
- El malware se ejecuta únicamente en sistemas configurados en idioma ruso.
- La carga útil se inserta en cada repositorio accesible de GitHub añadiendo un archivo «.claude/settings.json» que explota el hook SessionStart de Claude Code, así como un archivo «.vscode/tasks.json» con la configuración «runOn»: «folderOpen», provocando que cualquier intento de abrir el repositorio infectado en Visual Studio Code o Claude Code ejecute el malware.

«Este representa uno de los primeros ataques a la cadena de suministro que aprovecha configuraciones de agentes de codificación con IA como vector de persistencia y propagación,» afirmó StepSecurity.

Un análisis más profundo del origen del problema reveló que los atacantes comprometieron la cuenta de RoshniNaveenaS para los tres paquetes «@cap-js», luego subieron un flujo de trabajo modificado a una rama secundaria y utilizaron el token OIDC de npm obtenido para publicar los paquetes maliciosos sin trazabilidad. En el caso de mbt, se sospecha que hubo compromiso del token estático de npm «cloudmtabot» a través de un método aún no determinado.

«El equipo cds-dbs migró a un sistema de publicación confiable con OIDC de npm en noviembre de 2025,» explicó SafeDep. *«Bajo este esquema, GitHub Actions puede solicitar un token temporal de npm sin necesidad de almacenar secretos persistentes en el repositorio. El atacante replicó manualmente este proceso en un paso de CI y mostró el token resultante.»*

«La brecha crítica de configuración: la política de publicador confiable OIDC de npm para @cap-js/sqlite aceptaba cualquier flujo de trabajo dentro de cap-js/cds-dbs, no únicamente el archivo release-please.yml en la rama principal. Un push a cualquier rama podía intercambiar un token OIDC en nombre del paquete si el flujo contaba con permisos id-token: write y la referencia environment: npm.»

Como respuesta al incidente, los mantenedores han publicado nuevas versiones seguras que reemplazan las ediciones comprometidas:

- sqlite: v2.4.0, v2.3.0
- postgres: v2.3.0, v2.2.2
- hana: v2.8.0, v2.7.2
- db-service: v2.10.1
- mbt: v1.2.49