Expertos en ciberseguridad han identificado varios paquetes maliciosos en el registro de npm que imitan la herramienta Hardhat de la Fundación Nomic, con el objetivo de extraer datos confidenciales de los sistemas de los desarrolladores.

> *«Al aprovechar la confianza en los complementos de código abierto, los atacantes han introducido paquetes maliciosos en npm, permitiéndoles extraer información crítica como claves privadas, frases mnemotécnicas y configuraciones específicas»*, explicó el equipo de investigación de Socket en su análisis.

Hardhat es una plataforma de desarrollo para software basado en Ethereum, que integra herramientas para la edición, compilación, depuración y despliegue de contratos inteligentes y aplicaciones descentralizadas (dApps).

La lista de los paquetes identificados es la siguiente:

- nomicsfoundations
- @nomisfoundation/hardhat-configure
- installedpackagepublish
- @nomisfoundation/hardhat-config
- @monicfoundation/hardhat-config
- @nomicsfoundation/sdk-test
- @nomicsfoundation/hardhat-config
- @nomicsfoundation/web3-sdk
- @nomicsfoundation/sdk-test1
- @nomicfoundations/hardhat-config
- crypto-nodes-validator
- solana-validator
- node-validators
- hardhat-deploy-others
- hardhat-gas-optimizer
- solidity-comments-extractors

Entre los paquetes maliciosos identificados, el más destacado es @nomicsfoundation/sdk-test, que ha sido descargado 1,092 veces. Este paquete, publicado originalmente en octubre de 2023, está diseñado para capturar frases mnemotécnicas y claves privadas del entorno Hardhat, enviándolas posteriormente a un servidor controlado por los atacantes.

> *«El ataque comienza cuando el paquete comprometido es instalado. Estos paquetes explotan el entorno de ejecución de Hardhat mediante funciones como `hreInit()` y `hreConfig()`, que recopilan datos sensibles, incluidas claves privadas, frases mnemotécnicas y configuraciones»*, detalló la compañía.
>
> *«Los datos recolectados se transmiten a servidores gestionados por los atacantes mediante claves codificadas y direcciones de Ethereum para facilitar el proceso de exfiltración.»*

Este descubrimiento sigue a la reciente identificación de otro paquete malicioso en npm llamado ethereumvulncontracthandler, que se presenta como una biblioteca para detectar vulnerabilidades en contratos inteligentes de Ethereum, pero que en realidad incluye un mecanismo para instalar el malware Quasar RAT.

En meses recientes, se han detectado paquetes maliciosos adicionales en npm que utilizan contratos inteligentes de Ethereum para distribuir direcciones de servidores de comando y control (C2). Estas campañas convierten sistemas infectados en parte de una botnet basada en blockchain conocida como MisakaNetwork. La actividad ha sido atribuida a un actor de amenazas de habla rusa identificado como «_lain».

> *«El atacante resalta una vulnerabilidad en el ecosistema de npm: los paquetes suelen depender de múltiples bibliotecas externas, creando una estructura compleja similar a 'capas anidadas', lo que complica las revisiones de seguridad exhaustivas»*, indicó Socket.

> *«_lain admite explotar esta complejidad y el desorden en las dependencias del ecosistema npm, sabiendo que es poco viable que los desarrolladores analicen cada biblioteca y sus dependencias.»*

Además, se han descubierto bibliotecas fraudulentas en los ecosistemas npm, PyPI y RubyGems, que utilizan herramientas de prueba de seguridad de aplicaciones fuera de banda (OAST) como oastify.com y oast.fun para extraer información confidencial hacia servidores controlados por los atacantes.

Entre los paquetes identificados se incluyen:

- adobe-dcapi-web (npm): evita comprometer sistemas Windows, Linux y macOS localizados en Rusia, pero recopila información del sistema.
- monoliht (PyPI): diseñado para recolectar metadatos del sistema.
- chauuuyhhn, nosvemosssadfsd, holaaaaaafasdf (RubyGems): contienen scripts que transfieren información sensible mediante consultas DNS a un endpoint de oastify.com.

> *«Las herramientas y métodos originalmente creados para evaluaciones de seguridad ética están siendo reutilizados por actores malintencionados. Lo que antes se utilizaba para identificar vulnerabilidades en aplicaciones web ahora se emplea para robar información, establecer canales de comando y control (C2), y llevar a cabo ataques en varias etapas»*, señaló Kirill Boychenko, investigador de Socket.

Para reducir los riesgos asociados con estos paquetes, se recomienda a los desarrolladores de software validar la autenticidad de los paquetes, ser precavidos al escribir los nombres de estos, y revisar el código fuente antes de proceder con su instalación.