Se ha dado a conocer una vulnerabilidad de seguridad crítica en LangChain Core que podría ser explotada por un atacante para robar secretos sensibles e incluso manipular las respuestas de los modelos de lenguaje de gran tamaño (LLM) mediante inyección de prompts.

LangChain Core (es decir, langchain-core) es un paquete central de Python dentro del ecosistema LangChain que proporciona las interfaces fundamentales y abstracciones independientes del modelo para desarrollar aplicaciones basadas en LLM.

La falla, identificada como CVE-2025-68664, cuenta con una puntuación CVSS de 9.3 sobre 10.0. El investigador de seguridad Yarden Porat fue quien reportó el problema el 4 de diciembre de 2025, y la vulnerabilidad ha sido bautizada con el nombre de LangGrinch.

"Existe una vulnerabilidad de inyección de serialización en las funciones dumps() y dumpd() de LangChain", indicaron los mantenedores del proyecto en un aviso de seguridad. *"Estas funciones no escapan correctamente los diccionarios que contienen claves 'lc' al serializar diccionarios de formato libre."*

"La clave 'lc' es utilizada internamente por LangChain para marcar objetos serializados. Cuando los datos controlados por el usuario incluyen esta estructura de clave, se tratan como un objeto legítimo de LangChain durante la deserialización, en lugar de considerarse datos planos del usuario."

De acuerdo con Porat, investigador de Cyata, el núcleo del problema radica en que ambas funciones no neutralizan adecuadamente los diccionarios controlados por el usuario que contienen claves "lc". Dicho marcador representa objetos de LangChain dentro del formato interno de serialización del framework.

"Una vez que un atacante logra que un flujo de orquestación de LangChain serialice y luego deserialice contenido que incluye una clave 'lc', se puede instanciar un objeto arbitrario inseguro, lo que potencialmente activa múltiples rutas favorables para el atacante", explicó Porat.

Esto puede derivar en distintos escenarios, como la extracción de secretos desde variables de entorno cuando la deserialización se realiza con “secrets_from_env=True” (configuración que anteriormente estaba habilitada por defecto), la instanciación de clases dentro de espacios de nombres confiables preaprobados —como langchain_core, langchain y langchain_community— e incluso la posible ejecución de código arbitrario a través de plantillas Jinja2.

Además, este fallo de escape permite la inyección de estructuras de objetos de LangChain mediante campos controlados por el usuario, como metadata, additional_kwargs o response_metadata, utilizando técnicas de inyección de prompts.

El parche publicado por LangChain introduce valores predeterminados más restrictivos en las funciones load() y loads(), mediante un parámetro de lista blanca llamado “allowed_objects”, que permite definir qué clases pueden ser serializadas o deserializadas. Asimismo, las plantillas Jinja2 quedan bloqueadas por defecto y la opción “secrets_from_env” ahora se establece en “False”, deshabilitando la carga automática de secretos desde el entorno.

Las siguientes versiones de langchain-core se ven afectadas por CVE-2025-68664:

- >= 1.0.0, < 1.2.5 (corregido en 1.2.5)
- < 0.3.81 (corregido en 0.3.81)

Cabe destacar que existe una vulnerabilidad de inyección de serialización similar en LangChain.js, que también se origina por no escapar correctamente objetos con claves “lc”, lo que facilita la extracción de secretos y la inyección de prompts. Esta falla ha sido registrada como CVE-2025-68665, con una puntuación CVSS de 8.6.

Afecta a los siguientes paquetes npm:

- @langchain/core >= 1.0.0, < 1.1.8 (corregido en 1.1.8)
- @langchain/core < 0.3.80 (corregido en 0.3.80)
- langchain >= 1.0.0, < 1.2.3 (corregido en 1.2.3)

- langchain < 0.3.37 (corregido en 0.3.37)

Dada la gravedad de la vulnerabilidad, se recomienda encarecidamente a los usuarios actualizar a una versión corregida lo antes posible para garantizar una protección adecuada.

"El vector de ataque más común se da a través de campos de respuesta del LLM como additional_kwargs o response_metadata, los cuales pueden ser manipulados mediante inyección de prompts y luego serializados o deserializados durante operaciones en streaming", concluyó Porat. *"Este es un claro ejemplo de la intersección entre la IA y la seguridad clásica, donde muchas organizaciones se ven sorprendidas. La salida de un LLM debe considerarse una entrada no confiable."*