Investigadores en ciberseguridad han revelado una vulnerabilidad de alta gravedad en el editor de código impulsado por inteligencia artificial (IA) llamado Cursor, la cual podría permitir la ejecución remota de código.

La falla, identificada como CVE-2025-54136 (con una puntuación CVSS de 7.2), ha sido denominada *MCPoison* por Check Point Research, debido a que explota una peculiaridad en la forma en que el software gestiona las modificaciones en las configuraciones del servidor del Protocolo de Contexto de Modelo (MCP, por sus siglas en inglés).

"Una vulnerabilidad en Cursor AI permite a un atacante ejecutar código de forma remota y persistente al modificar un archivo de configuración MCP previamente confiable, ya sea dentro de un repositorio compartido de GitHub o editándolo localmente en la máquina de la víctima", indicó Cursor en un aviso publicado la semana pasada.

"Una vez que un colaborador aprueba un MCP aparentemente inofensivo, el atacante puede sustituirlo silenciosamente por un comando malicioso (por ejemplo, calc.exe) sin que se genere ninguna advertencia o solicitud adicional".

El MCP es un estándar abierto creado por Anthropic que permite a los modelos de lenguaje de gran escala (LLMs) interactuar con herramientas externas, datos y servicios de manera uniforme. Fue introducido por la empresa en noviembre de 2024.

Según Check Point, la vulnerabilidad CVE-2025-54136 se relaciona con el hecho de que un atacante puede modificar el comportamiento de una configuración MCP después de que esta ha sido aprobada por el usuario en Cursor. El ataque se desarrolla de la siguiente forma:

- Agregar una configuración MCP aparentemente legítima (`.cursor/rules/mcp.json`) en un repositorio compartido
- Esperar a que la víctima descargue el código y lo apruebe una vez en Cursor
- Sustituir la configuración MCP por una carga maliciosa, como la ejecución de un script o una puerta trasera
- Lograr ejecución persistente de código cada vez que la víctima abra Cursor

El problema esencial es que, una vez que una configuración es aprobada, Cursor la considera confiable de forma indefinida, incluso si ha sido alterada. La explotación exitosa de esta falla no solo introduce riesgos en la cadena de suministro, sino que también permite el robo de datos e información confidencial sin que los usuarios lo adviertan.

Tras un proceso de divulgación responsable el 16 de julio de 2025, Cursor resolvió el problema en la versión 1.3, lanzada a finales de ese mes, exigiendo ahora una aprobación del usuario cada vez que se modifica una entrada en el archivo de configuración MCP.

"La falla revela una debilidad crítica en el modelo de confianza de los entornos de desarrollo asistidos por IA, elevando el nivel de riesgo para los equipos que integran LLMs y automatización en sus flujos de trabajo", señaló Check Point.

Este desarrollo se presenta pocos días después de que Aim Labs, Backslash Security y HiddenLayer revelaran múltiples debilidades en la herramienta de IA que podrían haber sido aprovechadas para ejecutar código de forma remota y evadir mecanismos de protección basados en listas de denegación. Estas también fueron corregidas en la versión 1.3.

Los hallazgos coinciden con la creciente adopción de la IA en procesos empresariales, incluyendo la generación de código mediante LLMs, lo cual amplía la superficie de ataque a nuevos riesgos como ataques a la cadena de suministro de IA, generación de código inseguro, envenenamiento de modelos, inyecciones de instrucciones (prompt injection), alucinaciones, respuestas inapropiadas y filtraciones de datos. Algunos ejemplos relevantes incluyen:

- Una prueba realizada a más de 100 LLMs sobre su capacidad para escribir código en Java, Python, C# y JavaScript reveló que el 45% de los fragmentos generados fallaban en pruebas de seguridad, introduciendo vulnerabilidades incluidas en el Top 10 de OWASP. Java lideró con un 72% de fallos, seguido por C# (45%), JavaScript (43%) y Python (38%).
- Un ataque denominado *LegalPwn* demostró que es posible usar textos legales, términos de servicio o políticas de privacidad como vectores de inyección de

instrucciones, ocultando comandos maliciosos dentro de componentes legítimos pero poco vigilados, lo que puede hacer que el modelo clasifique código peligroso como seguro.

- El ataque *man-in-the-prompt* emplea una extensión maliciosa del navegador —sin permisos especiales— para abrir una nueva pestaña en segundo plano, iniciar un chatbot de IA e inyectar instrucciones dañinas para extraer información y comprometer la integridad del modelo. Esto se debe a que cualquier extensión con acceso al DOM puede interactuar directamente con las instrucciones del chatbot.
- La técnica de *jailbreak* llamada *Fallacy Failure* engaña al modelo con premisas lógicamente inválidas para inducirlo a generar respuestas restringidas, rompiendo sus propias reglas internas.
- *MAS hijacking* manipula el flujo de control en sistemas multiagente (MAS), permitiendo ejecutar código malicioso de manera transversal entre dominios, canales y topologías, aprovechando el comportamiento autónomo de los sistemas de IA.
- Una técnica conocida como *Poisoned GPT-Generated Unified Format (GGUF) Templates* ataca la fase de inferencia del modelo al incrustar instrucciones maliciosas dentro de archivos de plantilla de conversación. Al ejecutarse durante la inferencia, este enfoque logra evadir controles y pasar desapercibido. Dado que los archivos GGUF se distribuyen a través de plataformas como Hugging Face, este ataque aprovecha la confianza en la cadena de suministro para activarse.
- También es posible comprometer los entornos de entrenamiento de aprendizaje automático como MLFlow, Amazon SageMaker o Azure ML, afectando la confidencialidad, integridad y disponibilidad de los modelos, lo cual puede derivar en movimientos laterales, escalación de privilegios y robo o envenenamiento de datos y modelos.
- Un estudio realizado por Anthropic reveló que los LLMs pueden adquirir características ocultas durante la destilación, un fenómeno llamado *aprendizaje subliminal*, que hace que los modelos transmitan comportamientos no deseados a través de datos generados, aunque parezcan no relacionados, lo que puede provocar desalineaciones y conductas dañinas.

"A medida que los modelos de lenguaje se integran profundamente en flujos de trabajo

automatizados, copilotos empresariales y herramientas para desarrolladores, el riesgo asociado a estas técnicas de jailbreak se incrementa drásticamente”, advirtió *Dor Sarig*, de Pillar Security. *“Los jailbreaks modernos pueden propagarse a través de cadenas contextuales, infectando un componente de IA y provocando fallos lógicos en cascada en sistemas interconectados”*.

“Estos ataques demuestran que la seguridad en IA requiere un enfoque completamente nuevo, ya que pueden eludir protecciones tradicionales sin depender de fallos estructurales o vulnerabilidades catalogadas. La debilidad radica en el propio lenguaje y razonamiento que los modelos están diseñados para imitar”.