Cursor corrigió una vulnerabilidad crítica en su editor de código de IA que permitía a los hackers ejecutar comandos a través de inyección rápida

Investigadores en ciberseguridad han revelado una vulnerabilidad de alta gravedad ya corregida en Cursor, un popular editor de código impulsado por inteligencia artificial (IA), que podría permitir la ejecución remota de código (RCE).

El fallo, identificado como CVE-2025-54135 (con una puntuación CVSS de 8.6), fue solucionado en la versión 1.3 publicada el 29 de julio de 2025. Esta falla fue bautizada como *CurXecute* por Aim Labs, el mismo equipo que anteriormente dio a conocer *EchoLeak*.

"Cursor se ejecuta con privilegios de nivel desarrollador, y al vincularse con un servidor MCP que recolecta datos externos no confiables, esos datos pueden alterar el flujo de control del agente y explotar dichos privilegios", explicó el equipo de Aim Labs.

"Al introducir datos manipulados al agente mediante MCP, un atacante puede lograr una ejecución remota de código completa bajo los privilegios del usuario, lo que permite desde ataques de ransomware y robo de datos, hasta manipulación y alucinaciones de la IA".

En términos prácticos, la ejecución remota puede activarse mediante una única inyección de comandos hospedada externamente, que reescribe en silencio el archivo `~/.cursor/mcp.json` y ejecuta órdenes controladas por el atacante.

La vulnerabilidad guarda similitud con *EchoLeak*, ya que las herramientas expuestas por servidores MCP —utilizadas por los modelos de IA para interactuar con sistemas externos, como consultas a bases de datos o llamadas a APIs— pueden recibir datos no confiables que alteren el comportamiento esperado del agente.

Particularmente, Aim Security identificó que el archivo `mcp.json`, usado para configurar servidores MCP personalizados en Cursor, puede activar automáticamente cualquier entrada nueva (por ejemplo, añadir un servidor MCP de Slack) sin requerir confirmación.

Este modo de ejecución automática es especialmente peligroso, ya que permite la ejecución inmediata de una carga maliciosa inyectada por el atacante mediante un mensaje de Slack. La secuencia del ataque se desarrolla así:

1. El usuario añade un servidor MCP de Slack mediante la interfaz de Cursor.
2. El atacante publica un mensaje en un canal público de Slack con una carga de inyección de comandos.
3. La víctima abre un nuevo chat y solicita al agente de Cursor que resuma sus mensajes de Slack con una orden como: *"Utiliza herramientas de Slack para resumir mis mensajes"*.
4. El agente encuentra un mensaje diseñado para introducir comandos maliciosos, como modificar el archivo de configuración para añadir otro servidor MCP con instrucciones dañinas (por ejemplo, *«touch ~/<archivo_con_payload_RCE>»*).

"La raíz del problema es que las nuevas entradas en el archivo JSON global de MCP se ejecutan de forma automática", señaló Aim Security. *"Incluso si se rechaza la edición, la ejecución del código ya ocurrió"*.

Lo preocupante de este ataque es su simplicidad, pero también pone en evidencia cómo las herramientas asistidas por IA pueden abrir nuevas superficies de ataque al procesar contenido externo, como los servidores MCP de terceros.

"A medida que los agentes de IA conectan mundos externos, internos e interactivos, los modelos de seguridad deben asumir que los contextos externos pueden afectar la ejecución del agente — y es necesario monitorear cada paso", agregó la empresa.

La versión 1.3 de Cursor también aborda otro problema relacionado con el modo de ejecución automática, el cual puede eludir con facilidad los mecanismos de protección basados en listas de denegación mediante técnicas como codificación en Base64, scripts de shell o comillas que disfrazan comandos peligrosos (por ejemplo, *«e»cho bypass»*).

Tras la divulgación responsable por parte del equipo de BackSlash Research, Cursor optó por eliminar por completo el uso de listas de denegación para la ejecución automática, adoptando en su lugar una lista de permitidos (allowlist).

"No hay que confiar ciegamente en las soluciones de seguridad integradas que ofrecen las

plataformas de codificación con IA", afirmaron los investigadores Mustafa Naamneh y Micah Gold. *"La responsabilidad recae en las organizaciones usuarias para garantizar que los sistemas basados en agentes estén adecuadamente protegidos"*.

Esta revelación coincide con los hallazgos de HiddenLayer, que descubrió que la lista de denegación ineficaz de Cursor puede ser explotada al ocultar instrucciones maliciosas dentro de un archivo `README.md` en GitHub, lo que permite al atacante robar claves API, credenciales SSH e incluso ejecutar comandos prohibidos.

"Cuando la víctima visualizó el proyecto en GitHub, la inyección de comandos no era visible, y le pidió a Cursor que hiciera 'git clone' del proyecto y lo ayudara a configurarlo, algo común en sistemas IDE con agentes", detallaron los investigadores Kasimir Schulz, Kenneth Yeung y Tom Bonner.

"Pero al revisar el README para seguir las instrucciones, la inyección de comandos tomó el control del modelo de IA y lo obligó a usar la herramienta 'grep' para buscar claves en el espacio de trabajo del usuario, y luego exfiltrarlas con 'curl'".

HiddenLayer también identificó debilidades adicionales que permiten filtrar el prompt del sistema de Cursor al sobreescribir la URL base usada para las solicitudes a la API de OpenAI hacia un modelo con proxy, y exfiltrar claves SSH privadas del usuario combinando dos herramientas aparentemente inocuas: `read_file` y `create_diagram`, en lo que llamaron un "ataque de combinación de herramientas".

En esencia, esto consiste en insertar una inyección de comandos dentro del archivo `README.md` de GitHub, el cual Cursor analiza cuando el usuario le pide que resuma el archivo, ejecutando así el comando oculto.

La instrucción maliciosa utiliza `read_file` para acceder a las claves privadas SSH del usuario y luego emplea `create_diagram` para enviarlas a una URL controlada por el atacante en webhook.site. Todos estos fallos han sido corregidos por Cursor en la versión 1.3.

Estas noticias sobre vulnerabilidades en Cursor coinciden con un ataque diseñado por Tracebit contra *Gemini CLI*, una herramienta de línea de comandos de Google orientada a tareas de codificación, que aprovechaba una configuración por defecto para exfiltrar datos sensibles silenciosamente a un servidor del atacante mediante `curl`.

Al igual que en el caso de Cursor, el ataque requiere que la víctima (1) le pida a Gemini CLI interactuar con un repositorio de GitHub creado por el atacante que contiene una inyección indirecta en el archivo `GEMINI.md`, y (2) incluya un comando aparentemente inofensivo en una lista de permitidos, como `grep`.

"La inyección de comandos en estos elementos, combinada con serias deficiencias de validación y presentación dentro de Gemini CLI, puede dar lugar a ejecuciones de código arbitrarias indetectables", afirmó Sam Cox, fundador y CTO de Tracebit.

Para mitigar los riesgos, se recomienda a los usuarios de Gemini CLI actualizar a la versión 0.1.14, lanzada el 25 de julio de 2025.