VanguardIA — Ficha técnica

Guardián de contexto que retira instrucciones ocultas de las salidas de herramientas antes de que lleguen al LLM.

Equipo: Andrés Santiago Rodríguez Salazar, Juan Felipe León Barrero, Sebastián Felipe Angarita Rico, David Hortua Bayona · Universidad de los Andes

Track: 1. Seguridad de agentes

Repositorio	Contacto
https://github.com/AmogusAzul/guardian-contexto	sf.angarita@uniandes.edu.co

Resumen

Un agente con herramientas obedece las instrucciones ocultas en lo que lee. VanguardIA parte cada salida en trozos y retira los que parecen una inyección y se salen del contexto de la tarea, antes de que lleguen al LLM. Sobre 1.562 salidas detecta el 77,6 % de los ataques, frente al 51,6 % del detector DeBERTa, con la mitad de falsos positivos (11,3 % contra 21,9 %) y siete veces más rápido en CPU: 22 ms contra 171 ms.

1. La amenaza

En junio de 2025 un correo con instrucciones ocultas bastó para que Microsoft 365 Copilot filtrara información interna sin que nadie hiciera clic. Es inyección indirecta: el agente lee lo que devuelve una herramienta —un correo, un archivo, un servidor MCP— y obedece las órdenes escondidas ahí. OWASP la lista como riesgo número uno en aplicaciones con LLM, y los detectores actuales descartan la salida entera al sospechar: el agente falla la tarea legítima.

2. Lo que construimos

VanguardIA es un guardián de contexto entre la herramienta y el LLM: toda salida pasa por `check(texto, tarea)` antes de entrar al contexto. Su diferencia: sanea en vez de bloquear. Parte la salida en trozos de una a tres oraciones —también las claves de un JSON—, calcula el embedding de cada uno con MiniLM-L6 que corre en CPU y le asigna dos puntajes: cuánto se parece a una instrucción inyectada y cuánto se aleja del contexto de la tarea. Retira un trozo si su puntaje de ataque es alto, o si es medio y además está fuera de contexto; el resto pasa, y el agente conserva la información legítima. Devuelve el texto limpio más un registro de lo retirado, sin tokens ni otro LLM.



Figura 1. VanguardIA se sitúa entre la herramienta y el agente.

3. Cómo lo evaluamos

Medimos en dos niveles. El guardián solo, sobre un conjunto propio de 1.562 salidas de herramienta: 781 ataques y 781 legítimas, de fuentes que no se usaron para entrenar y sin ningún texto de AgentDojo. Contra el filtro de palabras clave y contra el detector DeBERTa, con el mismo conjunto y la misma máquina. Reportamos detección junto a falsos positivos. Aparte, una prueba de integración en AgentDojo con Qwen3 8B local en Ollama.

4. Resultados

Métrica	Sin defensa	Palabras clave	DeBERTa	VanguardIA
Detección de ataques (781)	0 %	16,1 %	51,6 %	**77,6 %**
Falsos positivos (781 legítimas)	0 %	0,0 %	21,9 %	**11,3 %**
Falsos positivos en salidas de herramienta (399)	0 %	0,0 %	11,0 %	**1,75 %**
Latencia mediana en CPU	–	<0,1 ms	171 ms	**22 ms**

Tabla 1. Conjunto propio de 1.562 salidas de herramienta (781 ataques, 781 legítimas); mismo conjunto y misma máquina para todas las defensas. Sin defensa, ningún ataque se detecta y ninguna salida legítima se toca.

Sin ningún filtro, los 781 ataques llegan intactos al agente; VanguardIA retira 606: 77,6 puntos de mejora neta, al costo de que el 11,3 % de las salidas legítimas pierda algún trozo. Falla en ataques directos clásicos, donde atrapa solo el 27,5 %. En AgentDojo (un caso) el ataque dejó de funcionar.

5. Límites y riesgo de doble uso

DeBERTa nos gana en ataques directos clásicos (100 % contra 27,5 % en secuestro de rol): el puntaje de contexto no ayuda cuando la orden va en tema. El 11,3 % de las salidas legítimas pierde algún trozo, aunque en salidas reales de herramienta baja a 1,75 %. La prueba en AgentDojo es un solo caso: muestra integración, no magnitud. No probamos atacantes adaptativos y solo evaluamos en inglés.

Referencias · Debenedetti et al. (2024). AgentDojo. NeurIPS D&B. arXiv:2406.13352 · Debenedetti et al. (2025). Defeating Prompt Injections by Design (CaMeL). arXiv:2503.18813 · Chen et al. (2025). Can Indirect Prompt Injection Attacks Be Detected and Removed? ACL. arXiv:2502.16580 · Reddy y Gujral (2025). EchoLeak (CVE-2025-32711). arXiv:2509.10540