



Seguridad y gobierno práctico de la IA

Cinco notas para poner a trabajar agentes de IA con controles que puedas demostrar

Los asistentes y agentes de IA se conectan al correo, a los datos de clientes y a los sistemas de producción más rápido de lo que las organizaciones deciden quién responde por ellos. Estas cinco notas siguen un flujo de IA desde su aprobación hasta su operación: quién decide, a qué puede acceder el agente, de qué depende, qué lo detiene en ejecución y cómo demostrar después que los controles funcionaron. Cada una se sostiene sola; leídas juntas, forman una secuencia que un MSP, MSSP o VAR puede recorrer con un cliente.

Telekomplex • telekomplex.net • octubre de 2026

Contenido

1. Antes de que un agente de IA actúe, decide quién responde por él
2. Un agente de IA necesita su propia identidad, y menos permisos
3. Un escaneo limpio es solo el punto de partida: cómo evaluar herramientas de IA y servidores MCP
4. Cuando un documento intenta dar órdenes: protección en capas para agentes de IA
5. ¿Puedes demostrar que los controles funcionaron? Operar la seguridad de IA después del lanzamiento

Los escenarios son recomendaciones ilustrativas, no informes de despliegues reales. Las referencias a fabricantes son ejemplos con fuente; las capacidades y la documentación cambian, y los enlaces se revisaron el 1 de octubre de 2026.

Antes de que un agente de IA actúe, decide quién responde por él

Las capacidades de un sistema de IA pueden crecer más rápido que las decisiones sobre su uso. Aprobarlo para una tarea no debería convertirse en silencio en una aprobación para todo lo que venga después.

Un equipo de ventas conecta un asistente de IA a su base de datos de clientes. El piloto funciona bien: resume conversaciones y sugiere seguimientos. Luego alguien activa el envío de correos. Otra persona le da acceso a la información de precios. En pocas semanas, un asistente de redacción se convirtió en un sistema que puede asumir compromisos con los clientes.

¿Quién aprobó ese cambio de responsabilidad?

Esta situación hipotética ilustra un problema de gobierno: las capacidades de un sistema de IA pueden crecer más rápido que las decisiones de la organización sobre su uso. Aprobarlo para una tarea no debería convertirse en silencio en una aprobación para cada capacidad futura.

El primer control es una descripción clara del uso previsto. Documenta el objetivo de negocio, las personas afectadas, la información involucrada y las acciones que el sistema puede realizar. Distingue entre producir una recomendación y ejecutar una transacción. Una propuesta de descuento en borrador y un descuento emitido automáticamente tienen consecuencias distintas.

Asigna responsabilidades que las personas puedan ejercer de verdad. Un responsable de negocio define los resultados aceptables y acepta el riesgo de negocio residual. Un responsable técnico mantiene la integración. Los especialistas en seguridad y privacidad evalúan la exposición que corresponda. Un responsable de operación puede suspender el servicio y coordinar la recuperación. Una misma persona puede tener varios roles, pero la etiqueta de responsable solo sirve si viene con autoridad, tiempo y una vía de escalamiento.

El Marco de Gestión de Riesgos de IA del NIST ofrece una estructura voluntaria para gestionar el riesgo de IA mediante cuatro funciones: gobernar, mapear, medir y gestionar. Estas funciones sostienen una gestión de riesgos continua; adoptar el marco no certifica que un sistema en particular sea seguro. [NIST AI RMF \[1\]](#)

Para la implementación, sugiero un registro de despliegue breve que acompañe al sistema durante todo su ciclo de vida. Debe identificar el propósito aprobado, la configuración del modelo y de las herramientas, los datos accesibles, las acciones prohibidas, las aprobaciones necesarias, los resultados de la evaluación, las limitaciones conocidas y la fecha de revisión. Registra quién puede aprobar un cambio importante, incluida una nueva fuente de datos o permisos más amplios para las herramientas.

Volvamos al asistente de ventas. Su aprobación inicial podría permitirle leer los registros de los clientes asignados y preparar borradores de correo. Enviar un mensaje externo requeriría una decisión aparte,

respaldada por pruebas de la selección de destinatarios, la exactitud de los hechos, el manejo de información confidencial y el escalamiento cuando el asistente no tiene evidencia suficiente.

El gobierno también va más allá de la ciberseguridad. Un sistema puede proteger credenciales y aun así dar consejos engañosos, tratar a los clientes de forma desigual o crear un servicio inaccesible. Revisa los efectos previsibles sobre las personas, la confiabilidad que exige la tarea y cómo alguien puede cuestionar o corregir un resultado.

La supervisión humana debe tener un propósito definido. Dale a quien revisa la acción propuesta, la información que la respalda, la incertidumbre y las consecuencias probables. Si quien revisa solo ve un botón de aprobar, la organización creó un punto de control sin darle los medios para decidir con información.

Un punto de partida práctico es un solo flujo de trabajo con consecuencias. Mapea sus decisiones, asigna responsables, define los límites de operación y especifica qué evidencia justificaría ampliarlo. Conecta ese registro con la gestión de cambios, para que agregar una herramienta o reemplazar un modelo dispare la revisión correspondiente.

La prueba del gobierno es si una organización puede explicar por qué un sistema de IA tiene permitido actuar, dentro de qué límites y quién puede intervenir cuando esos límites ya no sirven.

Antes de ampliar tu próximo piloto, pregúntate: ¿qué capacidad requeriría una nueva aprobación, y quién tiene la autoridad para darla?

Fuentes

[1] <https://www.nist.gov/itl/ai-risk-management-framework>

Un agente de IA necesita su propia identidad, y menos permisos

Un agente que usa las credenciales de un gerente tiene más autoridad de la que su tarea necesita, antes de que el modelo cometa cualquier error. Dale su propia identidad y un conjunto de permisos más acotado.

Imagina un agente de soporte al cliente que usa las credenciales de un gerente. Puede leer tickets, emitir reembolsos, exportar registros de clientes y cambiar la configuración de cuentas, porque el gerente puede hacer las cuatro cosas. La tarea real del agente es mucho más acotada: resumir un ticket y preparar una respuesta.

El problema existe antes de que el modelo cometa un error. La integración le otorgó más autoridad de la que la tarea requiere.

OWASP describe la agencia excesiva como un riesgo que surge cuando una aplicación de IA tiene demasiada funcionalidad, demasiados permisos o demasiada autonomía. Reducir esa exposición implica examinar lo que la aplicación puede ejecutar, no solo mejorar las instrucciones que se le dan al modelo.

[OWASP: Excessive Agency \[1\]](#)

Empieza por separar tres conceptos. La identidad establece qué actor de software hace una solicitud. La autorización determina si esa solicitud está permitida. La rendición de cuentas identifica a las personas y equipos responsables de operar el sistema. Asociar un agente a un responsable no significa que ese responsable haya aprobado personalmente cada acción.

Un buen diseño de acceso le da al agente una identidad propia y gestionada, y le otorga permisos para su flujo de trabajo específico. Cuando actúa en nombre de un usuario, conserva ese contexto de delegación sin copiar el conjunto completo de permisos del usuario. El acceso efectivo debe respetar tanto los límites de la aplicación como el alcance permitido al usuario.

En nuestro ejemplo de soporte, permitiría al agente leer el ticket asignado y crear un borrador interno. Las exportaciones de datos de clientes y los cambios de cuenta no estarían disponibles. Un flujo de reembolsos usaría una herramienta aparte, definida de forma acotada, con verificaciones en el servidor sobre el ticket, el cliente, el monto y la aprobación.

Esta distinción importa porque permitir el nombre de una herramienta no basta. Una operación permitida de "leer cliente" podría pedir el registro del cliente equivocado. Aplica los límites de recurso y de tenant en el servicio dueño de los datos, aunque un gateway también haga verificaciones.

Las credenciales deben gestionarse fuera de los prompts y de los archivos de configuración comunes, con un gestor de secretos adecuado o un mecanismo de identidad de carga de trabajo. Prefiere credenciales

de alcance limitado y corta duración cuando la integración lo permite. Una duración corta reduce la exposición, pero no garantiza por sí sola una revocación inmediata.

En la autorización MCP sobre HTTP, valida los tokens para el recurso al que están destinados. La guía de seguridad de MCP rechaza explícitamente el reenvío de tokens: aceptar un token destinado a otro servicio y reenviarlo hacia abajo rompe los límites de autorización. El acceso a los servicios posteriores necesita su propio esquema de autorización, bien diseñado. [Guía de seguridad de MCP \[2\]](#)

Los roles de cada producto también merecen precisión. Cisco describe Duo como soporte para la identidad y el ciclo de vida de los agentes, y Secure Access como el que aplica políticas de acceso detalladas a través de su arquitectura de gateway MCP. Son funciones complementarias; registrar una identidad no basta para tener un control completo. [Solution brief de Cisco \[3\]](#)

Por último, prueba quitar el acceso. Desactiva el agente, cambia el rol de su responsable y ejercita sesiones antiguas y credenciales en caché en un entorno controlado. Mide cuándo se detiene realmente el acceso, incluidos los trabajos en cola y las conexiones posteriores. Actualizar un directorio es un evento; una revocación exitosa es un resultado observado.

Un registro de acceso sencillo puede reunir el agente, el responsable, la acción permitida, el alcance de recursos, las condiciones, la duración de las credenciales, el requisito de aprobación y el mecanismo de revocación.

La pregunta que hay que hacerse es concreta: si este agente malinterpretara hoy su tarea, ¿qué le impediría hacer la infraestructura?

Fuentes

[1] <https://genai.owasp.org/llmrisk/llm062025-excessive-agency/>

[2] https://github.com/modelcontextprotocol/modelcontextprotocol/blob/main/docs/docs/2026-07-28/tutorials/security/security_best_practices.mdx

[3] <https://www.cisco.com/c/en/us/solutions/collateral/artificial-intelligence/security/zero-trust-agentic-ai-sb.html>

Un escaneo limpio es solo el punto de partida: cómo evaluar herramientas de IA y servidores MCP

Usar el Model Context Protocol no demuestra que se pueda confiar en un servidor, en quien lo mantiene ni en los sistemas que hay detrás. Aprueba una versión exacta para una tarea exacta, con evidencia.

Un equipo descubre un servidor MCP que promete un acceso cómodo a datos del negocio. La demostración convence, las descripciones de las herramientas suenan razonables y un escáner no reporta hallazgos. El equipo aprueba la conexión.

Dos semanas después, el servidor cambia.

¿Qué versión aprobó la organización, y qué comportamiento evaluó realmente?

El Model Context Protocol ofrece una forma de conectar aplicaciones de IA con capacidades como herramientas y fuentes de datos. Usar ese protocolo no establece, por sí solo, que se pueda confiar en el servidor, en quien lo mantiene ni en los sistemas que hay detrás.

Por eso, la evaluación de la cadena de suministro debería empezar por la procedencia y el propósito. Identifica quién mantiene un componente, dónde se ejecuta, cómo se actualiza, qué dependencias incluye y qué datos salen de la organización. En un servicio alojado, establece qué evidencia puede entregar el proveedor y qué cambios pueden observar o controlar los clientes.

El proceso de admisión que sugiero separa descripción, implementación y comportamiento. La descripción de una herramienta dice lo que afirma hacer. El código y la configuración, cuando se pueden revisar, muestran cómo funciona. Una ejecución controlada muestra qué pasa en condiciones elegidas. Ninguna de estas miradas reemplaza por completo a las otras.

El MCP Scanner de código abierto de Cisco ilustra cómo pueden contribuir varios métodos de evaluación. Sus opciones documentadas incluyen reglas YARA, análisis basado en LLM, la API de inspección de Cisco AI Defense y modos para examinar el contenido MCP expuesto, el código fuente disponible y las dependencias. La cobertura depende del modo elegido, de los analizadores configurados y del material accesible. Un escaneo solo de metadatos no puede establecer todo lo que podría hacer el código del servidor. [Cisco MCP Scanner \[1\]](#)

Trata los hallazgos como evidencia para una decisión. Que no haya hallazgos significa que las verificaciones configuradas no identificaron un problema en el material examinado. No establece que toda respuesta o ejecución futura de una herramienta vaya a ser segura.

Para un conector hipotético de registros de clientes, primero documentaría el uso permitido: obtener un conjunto limitado de campos de una cuenta asignada. Luego probaría si fallan las solicitudes sobre otra

cuenta, si el conector intenta conexiones de red inesperadas y si una actualización cambia sus permisos o el contenido que devuelve.

Ejecuta los componentes desconocidos en un entorno de evaluación aislado, con datos sintéticos y credenciales restringidas. Esto aplica también al propio proceso de evaluación: un modo del escáner que lanza un servidor local ejecuta ese servidor, mientras que un escaneo de metadatos sin conexión tiene otra exposición.

El registro de aprobación debe identificar la versión del artefacto o del servicio cuando exista, la configuración, el inventario de herramientas, la fecha de evaluación, los hallazgos, las limitaciones aceptadas, quién revisó y qué dispararía una nueva evaluación. En el software propio, fija las versiones revisadas y sigue las actualizaciones. En los servicios alojados que no se pueden fijar, usa la información contractual sobre cambios, el monitoreo y evaluaciones repetidas para gestionar esa incertidumbre.

Extiende la misma disciplina a modelos, fuentes de recuperación de información, instrucciones de los agentes y demás dependencias. Reemplazar un modelo puede cambiar el comportamiento del flujo aunque su API siga siendo compatible. Un repositorio de documentos puede introducir contenido nuevo sin que el modelo cambie en absoluto.

Para un primer despliegue, crea un catálogo aprobado pequeño, con responsables designados y motivos explícitos de admisión. Haz que retirar y revertir sean parte del proceso del catálogo, para que descubrir un problema produzca una respuesta concreta.

La pregunta útil va más allá de si un escáner pasó: ¿qué evidencia respalda usar esta capacidad exacta, para esta tarea, con este nivel de acceso, y qué cambio invalidaría esa decisión?

Fuentes

[1] <https://github.com/cisco-ai-defense/mcp-scanner>

Cuando un documento intenta dar órdenes: protección en capas para agentes de IA

La inyección indirecta de instrucciones llega dentro del contenido que un agente está procesando. El objetivo es que el contenido no confiable nunca adquiera autoridad operativa.

Un empleado le pide a un asistente de IA que resuma una propuesta de un proveedor. Dentro del documento, un atacante colocó instrucciones que le dicen al asistente que obtenga los precios internos y los envíe a un destino externo.

El empleado pidió un resumen. El documento intenta crear una tarea nueva.

Este es un ejemplo hipotético de inyección indirecta de instrucciones: las instrucciones llegan a través del contenido que el sistema de IA está procesando y no a través de la solicitud del usuario autorizado. OWASP identifica este riesgo y recomienda combinar controles como restricciones de privilegios, validación de salidas, filtrado y supervisión. Ningún filtro por sí solo debería tratarse como una solución completa. [OWASP: Prompt Injection \[1\]](#)

El objetivo práctico es impedir que el contenido no confiable adquiera autoridad operativa. Marcar un documento como no confiable puede ayudar al modelo a interpretarlo, pero una instrucción del tipo "ignora el contenido malicioso" no reemplaza el control sobre lo que la aplicación puede ejecutar.

En el flujo del proveedor, haría posible el resumen sin darle al asistente acceso a los precios internos ni a una herramienta de envío externo. Así, un proceso de resumen comprometido tendría menos formas de causar daño. Si otro flujo necesita esas capacidades, dale permisos separados y condiciones de aprobación explícitas.

La protección en ejecución debe seguir el camino real de los datos y las acciones. Inspecciona el material entrante antes de que llegue al contexto del modelo cuando ese límite necesite protección. Valida una llamada a herramientas propuesta antes de ejecutarla. Revisa los datos que devuelven las herramientas antes de pasarlos al modelo cuando la aplicación requiera esa salvaguarda. Revisa el contenido saliente antes de entregarlo a su destino.

La ubicación determina lo que un control puede lograr. Una inspección hecha después de que el modelo recibió información sensible no puede deshacer esa exposición. Un gateway no puede inspeccionar una conexión directa que lo esquivo. Dibuja las rutas, incluidas las herramientas locales, los trabajos en segundo plano y las APIs externas, antes de afirmar que tienes cobertura.

Piensa en una operación autorizada que crea el registro de un proveedor. La aplicación debería validar los tipos de campo, los valores permitidos, la pertenencia al tenant y la autoridad de quien llama. En un

mensaje externo, inspecciona el destinatario real y los adjuntos elegidos. Estas verificaciones limitan la ejecución aun cuando la acción propuesta suene razonable en lenguaje natural.

La inspección de contenido suma otra fuente de evidencia. Puede identificar sospechas de inyección, información sensible o violaciones de políticas, pero hay que evaluarla tanto por las amenazas que se le escapan como por los bloqueos innecesarios. Decide cómo se comporta el flujo si el servicio de inspección no está disponible. Ante una acción con consecuencias, puede ser preferible pausar para revisión que seguir sin el control necesario.

La aprobación humana debe quedar atada a una acción concreta. Muestra el destinatario, el contenido, los datos involucrados y el efecto esperado. Si esos detalles cambian después de la aprobación, exige una nueva decisión cuando el cambio afecta el riesgo. Una aprobación vaga para "continuar" protege poco frente a una operación sustancialmente distinta.

Cisco posiciona AI Defense como un componente de protección en ejecución; DefenseClaw se describe como una capa de aplicación de políticas y de evidencia, con integraciones de escaneo e inspección. Estas herramientas necesitan una cobertura y una validación de políticas propias de cada despliegue. Su presencia no garantiza que se bloquee cualquier ataque. [Arquitectura de Cisco](#) [2], [DefenseClaw](#) [3]

Una prueba útil es colocar una instrucción maliciosa simulada e inofensiva en un documento de prueba y observar el flujo completo. El éxito significa que la tarea prevista sigue siendo útil mientras el acceso o la ejecución no autorizados se impiden y quedan registrados.

El límite que importa es el punto en que la sugerencia de un modelo se convierte en una acción en el mundo real.

Fuentes

[1] <https://genai.owasp.org/llmrisk/llm01-prompt-injection/>

[2] <https://www.cisco.com/c/en/us/solutions/collateral/artificial-intelligence/security/zero-trust-agentic-ai-sb.html>

[3] <https://github.com/cisco-ai-defense/defenseclaw>

¿Puedes demostrar que los controles funcionaron?

Operar la seguridad de IA después del lanzamiento

Una solicitud bloqueada en un tablero no prueba la contención. Operar la seguridad de IA exige evidencia que conecte las decisiones con sus efectos.

Un agente de IA intenta exportar información de clientes. Un tablero muestra una solicitud bloqueada y el equipo cierra la alerta.

Antes de declarar contenido el incidente, quedan varias preguntas. ¿Fue el único intento? ¿Tuvo éxito una solicitud anterior? ¿Podría seguir ejecutándose un trabajo en segundo plano? ¿Qué sistema registró el resultado final?

Operar la seguridad de IA exige evidencia que conecte las decisiones con sus efectos.

La guía de respuesta a incidentes del NIST ubica la preparación, la detección, la respuesta y la recuperación dentro de una gestión continua del riesgo de ciberseguridad. Los sistemas de IA deberían sumarse a ese proceso operativo, con atención adicional a sus modelos, herramientas, fuentes de datos y acciones delegadas. [NIST SP 800-61 Rev. 3](#) ^[1]

Para un flujo con agentes, sugiero un registro de eventos que identifique al agente, el contexto de usuario o de servicio que corresponda, el responsable, la operación solicitada, el recurso objetivo, la versión de la política, la decisión y el resultado de la ejecución. Usa identificadores de correlación para unir la aplicación, el gateway, el servidor de herramientas y el servicio posterior.

Mantén visible la diferencia entre solicitud y resultado. "Permitido" significa que un punto de control autorizó una operación; no establece que la operación se haya completado. "Bloqueado" describe una intervención concreta; no establece que se haya bloqueado cada camino hacia el mismo resultado.

Recoge suficiente información para reconstruir los hechos relevantes sin convertir el sistema de auditoría en un nuevo depósito de secretos. Oculta o excluye las credenciales, restringe el acceso a los registros sensibles y define la retención según las necesidades operativas y las obligaciones aplicables. Registra acciones y resultados observables; no tomes como prueba de lo que pasó una explicación generada por el modelo.

Un ejercicio de incidente ilustrativo puede volver concretos estos requisitos. Un agente de prueba intenta exportar datos sintéticos de clientes a un destino no aprobado. El equipo debe identificar el intento, confirmar si salió algún dato, detener el flujo correspondiente y preservar la evidencia. Después debe demostrar que los reintentos, el trabajo en cola y las conexiones alternativas no pueden continuar la operación.

Desactivar la identidad de un agente es un paso de contención. Según la arquitectura, quienes responden también pueden necesitar revocar credenciales, terminar procesos, cancelar trabajos, restringir el acceso a la red o desactivar un conector. El runbook debe decir qué equipo controla cada mecanismo y cómo verificar su efecto.

La recuperación merece la misma atención. Revisa los cambios que el agente ya hizo, repara los registros afectados cuando corresponda, rota los secretos expuestos y corrige la debilidad de fondo. Reinicia con una configuración controlada después de que pasen las pruebas pertinentes. Registra la incertidumbre que queda, en lugar de convertir una investigación incompleta en una afirmación de seguridad total.

Arma un set de evaluación repetible que incluya trabajo legítimo además de usos indebidos. De lo contrario, un sistema que bloquea todas las solicitudes podría parecer seguro mientras es inútil en la operación. Repite las pruebas pertinentes después de cambios importantes en el modelo, las instrucciones, las herramientas, los permisos o la política de inspección.

Entre las mediciones útiles están la proporción de acciones relevantes con registros de auditoría completos, los usos indebidos de prueba que lograron un resultado prohibido, las tareas legítimas bloqueadas por error y el tiempo observado para contener un incidente controlado. Informa las condiciones de la prueba y el total sobre el que se mide. Un resultado sobre un set de prueba pequeño es evidencia sobre esa prueba, no una tasa de protección universal.

El énfasis documentado de DefenseClaw en una evidencia de auditoría duradera es un ejemplo de herramienta que puede apoyar este proceso. Las organizaciones igual necesitan conectar esa evidencia con los registros de identidad y los resultados en los sistemas posteriores. [Repositorio de DefenseClaw](#) [2]

La revisión mensual debería terminar en decisiones: acotar el alcance de un agente, reparar una integración, mejorar una alerta, aceptar un riesgo residual documentado o retirar una capacidad.

La pregunta operativa es simple: si un agente actuara fuera de su tarea esta tarde, ¿podría tu equipo establecer qué pasó, detener el daño y justificar que es seguro volver a operar?

Fuentes

[1] <https://csrc.nist.gov/pubs/sp/800/61/r3/final>

[2] <https://github.com/cisco-ai-defense/defenseclaw>

Entregado con tu marca

Telekomplex da a MSPs, MSSPs y VARs capacidad de ingeniería certificada para gobernar, proteger y operar la IA de sus clientes, desde el uso de IA por los empleados y la identidad de los agentes hasta la protección en ejecución y la operación de seguridad de IA, en marca blanca y dentro de tu modelo de entrega.

telekomplex.net/es/ai-security • contact@telekomplex.net