Centro de Respuestas a Incidentes Informáticos
CSIRT Académico UNAD

Protegiendo Aplicaciones de IA: Identificación y Mitigación de Riesgos en LLMs

[1] 2025 Top 10 Risk & Mitigations for LLMs and Gen AI Apps

VIEM Vicerrectoría de Innovación y Emprendimiento
ECBTI Escuela de Ciencias Básicas, Tecnología e Ingeniería
Semillero de Investigación Ceros y Unos

Boletín de Ciberseguridad

Medio de Divulgación del Centro de Respuestas a Incidentes Informáticos: CSIRT Académico UNAD

E-boletín Informativo CSIRT Académico UNAD

Edición electrónica, financiada por la Universidad Nacional Abierta y a Distancia (UNAD)

Medio de divulgación: Correo Electrónico, Sitio Web

Incluye: Noticias, alertas o informes relacionados con la disciplina de la ciberseguridad

Número treinta
Marzo de 2025

Universidad Nacional Abierta y a Distancia (UNAD)
Vicerrectoría de Innovación y Emprendimiento (VIEM)
Escuela de Ciencias Básicas Tecnología Ingeniería (ECBTI)
Maestría en Ciberseguridad
Especialización en Seguridad Informática
CSIRT Académico UNAD

Universidad Nacional Abierta y a Distancia
Calle 14 sur No. 14-23 |Bogotá D.C
Correo electrónico: csirt@unad.edu.co
Página web: https://csirt.unad.edu.co

Vicerrectoría de Innovación y Emprendimiento (VIEM)
Ing. Andrés Ernesto Salinas - Vicerrector

Escuela de Ciencias Básicas Tecnología e Ingeniería (ECBTI)
Ing. Claudio Camilo González Clavijo – Decano

Maestría en Ciberseguridad (ECBTI)
Ing. Sonia Ximena Moreno Molano –
Líder Programa de Maestría en Ciberseguridad

Semillero de Investigación Ceros y Unos, adscrito al Grupo de Byte InDesign

Centro de Respuestas a Incidentes Informáticos CSIRT Académico UNAD
Ing. Luis Fernando Zambrano Hernández – Líder CSIRT Académico UNAD

Responsable de la Edición
Ing. Luis Fernando Zambrano Hernandez

Revisó
Libardo Cárdenas Corral
Analista CSIRT Académico UNAD

Estado legal:
Periodicidad: Mensual
ISSN: 2806-0164

Tabla de Contenido

Boletín informativo Número 30 4
Introducción 4
Desarrollo 5
Arquitectura multi agente 6
Amenazas y mitigación 7
Taxonomía de amenazas 8
OWASP Top 10 para aplicaciones LLM 9
LLM01: Prompt Injection (inyección de indicaciones) 9
LLM02:2025 Sensitive Information Disclosure (Divulgación de información sensible) 9
LLM03:2025 Supply Chain: (Cadena de suministro) 10
LLM04: Data and Model Poisoning: (Envenenamiento de datos y modelos) 10
LLM05:2025 Improper Output Handling (Manejo incorrecto de salida) 11
LLM06:2025 Excessive Agency (Agencia excesiva) 11
LLM07:2025 System Prompt Leakage (Filtración de Prompts del Sistema) 12
LLM08:2025 Vector and Embedding Weaknesses (Debilidades en Vectores y Embeddings) 12
LLM09:2025 Misinformation (Desinformación) 13
LLM05:2025 Improper Output Handling (Consumo Ilimitado) 13
Conclusiones 14
Canales de comunicación 15
Referentes bibliográficos 16

Centro de Respuestas a Incidentes Informáticos
CSIRT Académico UNAD

Boletín de Ciberseguridad

Boletín informativo Número 30

Marzo 2025

Protegiendo Aplicaciones de IA: Identificación y Mitigación de Riesgos en LLMs

Autores:

Luis Fernando Zambrano Hernández
CSIRT Académico UNAD
https://orcid.org/0000-0002-4690-3526

Hernando José Peña Hidalgo
CSIRT Académico UNAD
https://orcid.org/0000-0002-3477-2645

Sonia Ximena Moreno Molano
Líder Maestría en Ciberseguridad
https://orcid.org/0009-0002-6133-5157

Néstor Raúl Cárdenas Corral
CSIRT Académico UNAD
https://orcid.org/0000-0003-3691-0148

Introducción

Ilustración 1. Obtenida de https://owasp.org.

En la era digital actual, los Modelos de Lenguaje de Gran Escala (LLMs, por sus siglas en inglés) se han convertido en herramientas fundamentales para diversas aplicaciones, desde asistentes virtuales hasta sistemas de recomendación.

Sin embargo, su creciente adopción ha traído consigo una serie de riesgos de seguridad que requieren atención especializada.

El proyecto OWASP Top 10 para Aplicaciones de LLM que es una iniciativa comunitaria destacada en el ámbito de la seguridad informática, la cual ha identificado y clasificado las diez vulnerabilidades más críticas que afectan a las aplicaciones basadas en LLMs. Entre estas vulnerabilidades se encuentran la inyección de prompts, la divulgación de información sensible, las vulnerabilidades en la cadena de suministro, el envenenamiento de datos y modelos, y el manejo incorrecto de salidas, entre otras (Oligio, 2025).

Este boletín tiene como objetivo abordar cada una de estas vulnerabilidades, proporcionando ejemplos concretos y estrategias de mitigación mínimas, teniendo como propósito, además, equipar a desarrolladores, arquitectos y profesionales de la seguridad con el conocimiento necesario para proteger las aplicaciones de IA en un panorama tecnológico en constante evolución.

Desarrollo

La infografía "*Agentic AI – Threats and Mitigations*[1]" (OWASP, 2025) trata sobre las amenazas de seguridad en sistemas de inteligencia artificial (IA) con agentes autónomos[2]. Forma parte del proyecto de OWASP (Open Web Application Security Project) sobre amenazas en IA, centrándose en los riesgos de los sistemas que ejecutan acciones de manera autónoma utilizando herramientas, comandos del sistema o integraciones externas

Agencia y Razonamiento: Se enfoca en cómo un agente de IA determina de manera independiente los pasos necesarios para alcanzar sus objetivos.

✓ Indica, que ciertos riesgos de seguridad pueden surgir debido a esta autonomía

Memoria y Contexto: Se centra en cómo un agente de IA utiliza la memoria almacenada para tomar decisiones.

✓ Así mismo indica que este proceso puede ser vulnerable a manipulaciones y errores.

Herramientas y Ejecución: Analiza cómo los agentes de IA pueden interactuar con herramientas externas y ejecutar comandos del sistema.

✓ Resalta las amenazas de seguridad derivadas de estas acciones.

Identidad y Autenticación: Analiza si el sistema de IA depende de la autenticación para verificar la identidad de usuarios, herramientas o servicios.

✓ Destaca los riesgos de suplantación de identidad y falsificación.

Interacción Humana: Se enfoca en si la IA requiere de la intervención humana para operar eficazmente.

✓ Destaca que la relación entre humanos e IA puede presentar riesgos si no se gestiona adecuadamente.

Multi agente: Se enfoca en si un sistema de IA depende de múltiples agentes que interactúan entre sí.

✓ Resalta que este tipo de interacciones puede ser vulnerable a amenazas de seguridad

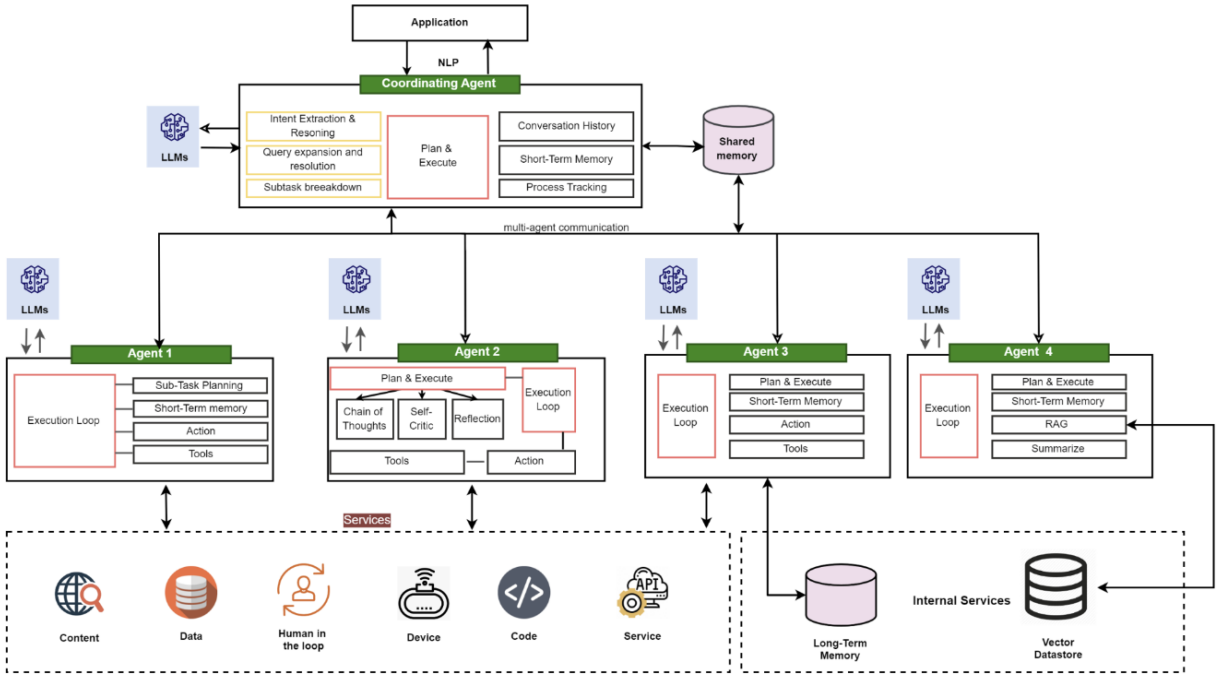
[1] Agentic: se refiere a sistemas de Inteligencia Artificial (IA) para el desarrollo automático de procesos, es decir, aquellos que pueden operar de manera autónoma, tomar decisiones y ejecutar acciones sin intervención humana directa.

[2] https://genai.owasp.org/resource/owasp-gen-ai-security-project-agentic-threats-navigator/

Arquitectura multi agente

Una arquitectura multi agente está compuesta por diversos agentes que pueden desempeñar funciones especializadas y escalar según sea necesario dentro de un sistema autónomo. La estructura general de la arquitectura se mantiene similar en la mayoría de los casos, salvo por la posible comunicación entre agentes y la inclusión opcional de un agente coordinador que coordina las interacciones. El diseño de una solución puede incorporar agentes especializados con funcionalidades avanzadas, dependiendo de los requisitos del sistema. A continuación, se ilustra una arquitectura multi agente con roles y capacidades especializadas (OWASP, Agentic AI – Threats and Mitigations, 2025).

Ilustración 2. Arquitectura multi agente con roles y capacidades especializadas



Fuente. https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/ (Pág. 11)

La ilustración anterior, representa un modelo de arquitectura de sistemas de agentes de IA, basado en el documento Agentic-AI-Threats-and-Mitigations_v1.0.1. En este esquema, múltiples agentes trabajan de manera coordinada utilizando Modelos de Lenguaje de Gran Escala (LLMs), memoria compartida y servicios externos donde se denota la siguiente lógica:

Coordinating Agent (Agente Coordinador): Funciona como el cerebro central que dirige las operaciones de los demás agentes. Este realiza

- La extracción de intención y razonamiento (Intent Extraction & Reasoning).

- La expansión de consultas y resolución de subtareas.
- La gestión de memoria a corto plazo y seguimiento de procesos.
- Este se comunica con memoria compartida para almacenar y recuperar información en tiempo real.

Agentes Especializados: Cada agente tiene una función específica, pero todos siguen un modelo de Planificación y Ejecución (Plan & Execute) en este sentido todos los agentes se apoyan en LLMs, lo que los hace susceptibles a riesgos como alucinaciones, envenenamiento de memoria y manipulación de datos.

Funciones de cada agente:

- Agent 1 → Planificación de subtareas, memoria a corto plazo y herramientas.
- Agent 2 → Utiliza Chain of Thoughts, Autoevaluación (Self-Critic) y Reflexión para mejorar decisiones.
- Agent 3 → Ejecuta tareas con planificación, memoria a corto plazo y herramientas.
- Agent 4 → Se especializa en Recuperación Aumentada por Generación (RAG) y Resumen de información.

Servicios Externos e Internos: Los agentes realizan proceso de comunicación con:

- Servicios externos (Services): Contenido, Datos, Humanos en el ciclo (HITL), Dispositivos, Código, APIs.
- Servicios internos (Internal Services): Memoria a Largo Plazo y Base de Datos Vectorial (Vector Datastore).

Amenazas y mitigación

El documento "*Agentic AI - Threats and Mitigations v1.0.1*[3]" es un informe detallado de la Iniciativa de Seguridad de IA de OWASP. Se centra en los riesgos de seguridad asociados con sistemas de IA autónomos, especialmente aquellos impulsados por Modelos de Lenguaje de Gran Escala (LLMs) y generativos AI. Algunos temas claves de este documento son:

Introducción a los agentes en IA	Arquitectura de referencia de agentes IA	Modelo de Amenazas en agentes de IA
Explica el concepto de los agentes, que se refiere a sistemas de IA capaces de razonar, tomar decisiones y ejecutar acciones de forma autónoma. Así mismo describe cómo la integración con modelos de lenguaje ha ampliado sus capacidades y sus riesgos.	Presenta la estructura típica de estos sistemas, incluyendo memoria, herramientas de ejecución y modelos de toma de decisiones. También explica cómo los agentes pueden interactuar entre sí en entornos multi agente.	Utiliza metodologías de modelado de amenazas para identificar riesgos específicos de sistemas de IA autónomos. Introduce el navegador de amenazas del agente de IA, una herramienta estructurada para evaluar riesgos

[3] https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

Taxonomía de amenazas

En el contexto de los sistemas de agentes impulsados por inteligencia artificial, surgen diversas amenazas que pueden comprometer su seguridad, confiabilidad y desempeño. Estas amenazas se derivan del uso indebido de herramientas, la manipulación de datos, la explotación de vulnerabilidades y la influencia de tácticas engañosas. Entre los riesgos más relevantes se encuentran el *poisoning* de memoria, el uso indebido de herramientas, el compromiso de privilegios, las alucinaciones en cascada y la suplantación de identidad.

Ilustración 3. Amenazas presentes a partir del uso de atentes de IA

Elaborado con https://app.napkin.ai/

El **poisoning** de memoria se refiere a la corrupción de la memoria del agente, lo que puede alterar su comportamiento esperado. Esta amenaza puede surgir cuando datos maliciosos son introducidos en el sistema, lo que lleva a decisiones erróneas o a un funcionamiento ineficiente. Los atacantes pueden manipular la información que el agente utiliza para aprender y tomar decisiones, afectando así su rendimiento y fiabilidad.
El uso **indebido** de herramientas implica la manipulación del agente para abusar de herramientas externas. Esto puede incluir la explotación de APIs o interfaces que el agente utiliza para interactuar con otros sistemas. Los atacantes pueden aprovechar vulnerabilidades en estas herramientas para ejecutar acciones no autorizadas o para obtener acceso a información sensible, comprometiendo la seguridad del sistema
El **compromiso** de privilegios se refiere a la elevación de permisos no autorizada dentro del sistema. Un atacante puede intentar obtener acceso a funciones o datos que normalmente están restringidos, lo que puede resultar en un control total sobre el agente o en la capacidad de realizar acciones dañinas. Esta amenaza es particularmente crítica en sistemas donde los permisos son mal gestionados o donde existen vulnerabilidades en la autenticación.
Las **alucinaciones** en cascada son errores generados por el agente que se amplifican con el tiempo. Estos errores pueden surgir de decisiones incorrectas que el agente toma debido a datos erróneos o mal interpretados. A medida que el agente continúa operando, estos errores pueden propagarse y llevar a resultados cada vez más distorsionados, afectando la calidad de las decisiones y la confianza en el sistema.
La **suplantación** de identidad y la manipulación humana implican engaños diseñados para que el agente tome decisiones erróneas. Esto puede incluir técnicas de ingeniería social que inducen al agente a actuar de manera que favorezca al atacante. La manipulación puede ser sutil, utilizando información engañosa o creando situaciones que lleven al agente a cometer errores críticos.

Elaboración propia

Nota.* Cada una de estas amenazas representa un desafío para la integridad de los agentes, afectando su capacidad para tomar decisiones precisas y seguras. La detección y mitigación de estos riesgos es fundamental para garantizar la correcta operación de los sistemas inteligentes en entornos críticos

OWASP Top 10 para aplicaciones LLM

El OWASP Top 10 para Aplicaciones LLM 2025[4] (OWASP, OWASP Top 10 for LLM Applications 2025, 2025) es una guía esencial que identifica las principales vulnerabilidades y riesgos de seguridad asociados con el uso de estos modelos. Desde ataques de inyección de prompts hasta filtración de información sensible y desinformación generada por IA. A medida que estas aplicaciones evolucionan, también lo hacen los riesgos, lo que hace imprescindible contar con un enfoque sólido y actualizado para proteger los sistemas basados en LLM.

LLM01: Prompt Injection (inyección de indicaciones)

Descripción del ataque	Riesgos	Ejemplo
Un atacante manipula los prompts ingresados en un LLM para alterar su comportamiento y obtener respuestas no deseadas. Esto puede ocurrir de forma directa, cuando el usuario introduce instrucciones maliciosas, o indirecta, cuando el modelo procesa datos externos que contienen comandos ocultos.	Filtración de información sensible (datos privados o empresariales). Ejecución de comandos no autorizados en sistemas conectados. Manipulación del contenido, generando información falsa o sesgada. Ataques multimodales, donde las instrucciones ocultas en imágenes o texto combinados amplían la superficie de ataque.	Un atacante inyecta un prompt en un chatbot de servicio al cliente, logrando que ignore sus reglas y acceda a bases de datos privadas, enviando información confidencial por correo electrónico.

Mitigación:

- ✓ Restringir el comportamiento del modelo, definiendo instrucciones estrictas en los prompts del sistema.
- ✓ Filtrar y validar entradas y salidas, bloqueando contenido malicioso.
- ✓ Aplicar el principio de mínimo privilegio, evitando que el modelo acceda a funciones críticas.
- ✓ Separar contenido confiable de fuentes externas para evitar inyecciones indirectas.
- ✓ Realizar auditorías y pruebas adversariales, simulando ataques para evaluar la seguridad del sistema.

LLM02:2025 Sensitive Information Disclosure (Divulgación de información sensible)

Descripción del ataque	Riesgos	Ejemplo
Ocurre cuando un LLM expone datos confidenciales en sus respuestas. Esto incluye información personal identificable (PII), detalles financieros, registros de salud, datos empresariales, credenciales de seguridad y documentos legales.	Filtración de datos personales (PII) en respuestas generadas por el modelo. Exposición de algoritmos y datos propietarios, facilitando ataques de inversión. Incorporación accidental de información confidencial en respuestas debido a entrenamiento descuidado.	Un usuario interactúa con un chatbot de servicio al cliente y recibe información privada de otro cliente debido a una falta de sanitización en los datos de entrenamiento, exponiendo detalles financieros y personales sin autorización.

Mitigación:

- ✓ Sanitización de datos: Implementar técnicas para filtrar información sensible antes de su uso en el modelo.
- ✓ Validación robusta de entradas: Detectar y bloquear datos confidenciales antes de que sean procesados.
- ✓ Control de acceso estricto: Aplicar el principio de mínimo privilegio, restringiendo el acceso solo a usuarios autorizados.
- ✓ Aprendizaje federado y privacidad diferencial: Reducir la exposición de datos centralizados y agregar ruido a los resultados para proteger la privacidad.

[4] https://owasp.org/www-project-top-10-for-large-language-model-applications/

LLM03:2025 Supply Chain: (Cadena de suministro)

Descripción del ataque	Riesgos	Ejemplo
La seguridad de la cadena de suministro de LLMs puede verse comprometida por modelos pre-entrenados, dependencias de terceros y plataformas de desarrollo colaborativo. Los atacantes pueden modificar o envenenar modelos, explotar vulnerabilidades en bibliotecas externas o infiltrarse en procesos de ajuste fino, afectando la integridad y seguridad de las aplicaciones de IA.	Uso de componentes obsoletos o vulnerables que pueden ser explotados. Falta de trazabilidad de los modelos, permitiendo que se usen versiones manipuladas. Alteración maliciosa de modelos pre-entrenados, introduciendo puertas traseras o sesgos. Explotación de entornos colaborativos, permitiendo la inserción de código malicioso.	Un atacante sube una versión comprometida de un modelo popular en Hugging Face. Los desarrolladores confían en su autenticidad y lo integran en sus aplicaciones, permitiendo la ejecución de código malicioso y la manipulación de respuestas generadas.

Mitigación:

- ✓ Verificar fuentes y proveedores antes de usar modelos o datos externos.
- ✓ Escanear vulnerabilidades y aplicar parches en dependencias críticas.
- ✓ Utilizar modelos con firmas digitales y validaciones de integridad.
- ✓ Realizar auditorías de seguridad en entornos de desarrollo colaborativo.
- ✓ Monitorear modelos para detectar anomalías y prevenir envenenamiento de datos.

LLM04: Data and Model Poisoning: (Envenenamiento de datos y modelos)

Descripción del ataque	Riesgos	Ejemplo
El envenenamiento de datos ocurre cuando los atacantes manipulan los datos utilizados en el preentrenamiento, ajuste fino o embeddings[5] de un LLM para introducir vulnerabilidades, sesgos o puertas traseras ocultas. Esto puede hacer que el modelo genere respuestas incorrectas, maliciosas o sesgadas, afectando su seguridad, precisión y comportamiento ético.	Generación de contenido sesgado o tóxico, afectando la credibilidad del modelo. Inserción de puertas traseras que pueden activarse con disparadores ocultos. Exposición de información sensible, si los datos inyectados provienen de fuentes externas no verificadas.	Un atacante inyecta datos manipulados en un modelo de lenguaje, logrando que responda con desinformación de manera intencionada. Usuarios que confían en el modelo reciben información falsa, afectando decisiones empresariales o políticas.

Mitigación:

- ✓ Verificar el origen de los datos con herramientas como OWASP CycloneDX.
- ✓ Evaluar proveedores de datos y contrastar resultados con fuentes confiables.
- ✓ Implementar detección de anomalías para filtrar datos envenenados antes del entrenamiento.
- ✓ Monitorear el comportamiento del modelo para identificar signos de manipulación.
- ✓ Utilizar control de versiones de datos (DVC) para detectar alteraciones en conjuntos de entrenamiento.

[5] **Embeddings**: técnica de procesamiento de lenguaje natural que convierte el lenguaje humano en vectores matemáticos Recuperador de: https://gustavo-espindola.medium.com/qu%C3%A9-son-los-embeddings-y-c%C3%B3mo-se-utilizan-en-la-inteligencia-artificial-con-python-45b751ed86a5

LLM05:2025 Improper Output Handling (Manejo incorrecto de salida)

Descripción del ataque	Riesgos	Ejemplo
Ocurre cuando las respuestas generadas por un LLM no son validadas, sanitizadas o procesadas adecuadamente antes de ser utilizadas en otros sistemas. Esto puede llevar a la ejecución de código malicioso, ataques de inyección o escalación de privilegios. Dado que los LLM pueden generar contenido basado en *prompts*, esta vulnerabilidad puede equivaler a proporcionar acceso indirecto a funcionalidades del sistema.	Ejecución remota de código (RCE) si la salida del LLM se usa en funciones como exec o eval. Ataques de Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF) en navegadores web. Inyección de SQL si las consultas generadas por el LLM no están parametrizadas. Vulnerabilidades de *path traversal* si los LLM construyen rutas de archivos sin sanitización. Ataques de phishing si el contenido generado por el LLM se usa en plantillas de correo sin escapado adecuado.	Un LLM genera dinámicamente consultas SQL basadas en las solicitudes de los usuarios. Si las entradas no se validan ni parametrizan, un usuario malintencionado podría solicitar una consulta que elimine todas las tablas de la base de datos, causando una pérdida de datos masiva.

Mitigación:

- ✓ Adoptar un enfoque de *zero-trust*, tratando la salida del modelo como la de un usuario desconocido.
- ✓ Validar y sanitizar todas las salidas del LLM antes de que sean utilizadas en otros sistemas.
- ✓ Codificar correctamente la salida según el contexto, usando HTML encoding, SQL escaping, y reglas de sanitización.
- ✓ Utilizar consultas parametrizadas en interacciones con bases de datos para prevenir SQL injection.
- ✓ Aplicar políticas estrictas de seguridad de contenido (CSP) para evitar ataques XSS en navegadores.
- ✓ Implementar monitoreo y registro de actividad para detectar patrones inusuales en las respuestas del LLM.

LLM06:2025 Excessive Agency (Agentica excesiva)

Descripción del ataque	Riesgos	Ejemplo
Ocurre cuando un LLM tiene demasiadas capacidades o permisos para interactuar con otros sistemas mediante extensiones, herramientas o plugins. En algunos casos, el modelo puede decidir de manera autónoma qué funciones invocar, lo que aumenta el riesgo de realizar acciones no deseadas o peligrosas debido a salidas erróneas, manipulaciones o inyecciones de prompts.	Acceso no autorizado a funciones sensibles, como modificar o eliminar datos. Uso de permisos excesivos, lo que permite acciones más allá de las necesarias. Automatización de tareas sin supervisión, que podría llevar a la ejecución de comandos peligrosos. Interacción con sistemas externos comprometidos, ampliando la superficie de ataque.	Un asistente personal basado en LLM tiene acceso al correo electrónico del usuario para resumir mensajes. Sin embargo, el plugin utilizado también tiene la capacidad de enviar correos. Un atacante envía un correo malicioso con instrucciones ocultas, logrando que el LLM extraiga información sensible de la bandeja de entrada y la reenvíe automáticamente al atacante

Mitigación:

- ✓ Minimizar el uso de extensiones, permitiendo solo las estrictamente necesarias.
- ✓ Reducir las funcionalidades de los plugins, evitando comandos abiertos o excesivos.
- ✓ Restringir permisos en los sistemas externos, aplicando el principio de mínimo privilegio.
- ✓ Ejecutar acciones en el contexto del usuario, asegurando autenticación con OAuth y permisos limitados.
- ✓ Requerir aprobación humana para tareas de alto impacto.
- ✓ Monitorear y registrar la actividad de las extensiones para detectar anomalías.
- ✓ Implementar límites de tasa (*rate-limiting*) para reducir el daño en caso de explotación.

LLM07:2025 System Prompt Leakage (Filtración de Prompts del Sistema)

Descripción del ataque	Riesgos	Ejemplo
Ocurre cuando las instrucciones internas diseñadas para guiar el comportamiento de un LLM son expuestas a los usuarios, revelando información sensible. Esto puede incluir credenciales, reglas internas, permisos, criterios de filtrado o detalles sobre la arquitectura del sistema. Si un atacante accede a estos prompts, puede utilizarlos para burlar restricciones, ejecutar ataques de inyección de prompts o realizar escalación de privilegios.	Exposición de credenciales y claves de API, permitiendo accesos no autorizados. Divulgación de reglas internas, que facilita la manipulación de decisiones del sistema. Revelación de criterios de filtrado, permitiendo a atacantes diseñar estrategias para eludir restricciones. Identificación de estructuras de permisos, lo que podría derivar en ataques de escalación de privilegios.	Un LLM tiene un prompt del sistema que contiene credenciales para conectarse a una base de datos. Si el prompt se filtra a un atacante, este puede utilizar las credenciales para acceder a la base de datos de la aplicación y extraer información confidencial.

Mitigación:

- ✓ No incluir datos sensibles en los prompts del sistema, como claves de autenticación o estructuras de permisos.
- ✓ No confiar en los prompts del sistema para controlar comportamientos críticos, usando sistemas externos para validar accesos y restricciones.
- ✓ Implementar medidas de seguridad externas, como validación de salida, para evitar que el LLM revele información protegida.
- ✓ Separar privilegios correctamente, asegurando que distintos agentes tengan solo los permisos mínimos necesarios.
- ✓ Monitorear intentos de extracción de prompts y bloquear solicitudes sospechosas mediante auditoría y detección de anomalías.

LLM08:2025 Vector and Embedding Weaknesses (Debilidades en Vectores y Embeddings)

Descripción del ataque	Riesgos	Ejemplo
Las vulnerabilidades en vectores y embeddings representan un riesgo significativo en sistemas que utilizan Retrieval Augmented Generation (RAG) con modelos de lenguaje grande (LLM). Si los vectores y embeddings son generados, almacenados o recuperados de forma insegura.	Filtración de datos confidenciales, si los controles de acceso a embeddings son deficientes. Conflictos de información en entornos multiusuario, provocando fugas de datos entre distintos grupos de usuarios. Alteración del comportamiento del modelo, afectando su precisión o reduciendo características como la empatía en respuestas.	Un atacante envía un currículum con texto oculto que instruye al LLM a recomendar a un candidato sin considerar su calificación real. Cuando el sistema analiza el documento con RAG, sigue las instrucciones ocultas y recomienda al candidato de forma automática, manipulando el proceso de selección.

Mitigación:

- ✓ Implementar controles de acceso estrictos para embeddings y vectores.
- ✓ Validar y autenticar las fuentes de datos, asegurando que provienen de entidades confiables.
- ✓ Revisar combinaciones de datos, clasificando y etiquetando información para evitar conflictos de acceso.
- ✓ Monitorear y registrar actividades en bases de datos de vectores para detectar comportamientos sospechosos.
- ✓ Evaluar el impacto de RAG en el modelo, ajustando el proceso de aumento de datos para mantener cualidades esenciales como la empatía y precisión en las respuestas.

LLM09:2025 Misinformation (Desinformación)

Descripción del ataque	Riesgos	Ejemplo
Representa una vulnerabilidad crítica en aplicaciones que dependen de estos modelos. Ocurre cuando los LLMs generan información falsa o engañosa, pero que aparenta ser creíble. Una de las principales causas es la alucinación, en la que el modelo produce contenido sin fundamento basándose únicamente en patrones estadísticos. Además, los sesgos en los datos de entrenamiento y la falta de información completa pueden contribuir a este problema.	Errores en decisiones basadas en información incorrecta, lo que puede generar problemas legales o de seguridad. Difusión de afirmaciones sin respaldo, especialmente en áreas sensibles como salud y derecho. Falsa representación de experiencia, donde el modelo aparenta comprender temas complejos sin realmente hacerlo. Generación de código inseguro o inexistente, lo que puede introducir vulnerabilidades en sistemas de software.	Un atacante detecta nombres de paquetes de software falsos generados por un asistente de codificación basado en LLM. Luego, publica paquetes maliciosos con esos nombres en repositorios de código. Los desarrolladores confían en las recomendaciones del asistente, descargan los paquetes maliciosos e introducen vulnerabilidades en sus sistemas.

Mitigación:

- ✓ Usar Retrieval-Augmented Generation (RAG) para mejorar la precisión, recuperando información verificada en tiempo real.
- ✓ Ajuste fino del modelo para reducir errores, aplicando técnicas como *parameter-efficient tuning (PET)* y *chain-of-thought prompting*.
- ✓ Revisión humana y verificación cruzada con fuentes confiables antes de tomar decisiones basadas en el modelo.
- ✓ Mecanismos automáticos de validación, especialmente en entornos críticos.
- ✓ Comunicación de riesgos, informando a los usuarios sobre las posibles limitaciones y errores del modelo.
- ✓ Buenas prácticas de codificación segura, evitando la integración de código incorrecto o vulnerable.
- ✓ Diseño de interfaces responsables, agregando filtros de contenido y etiquetas que adviertan sobre la confiabilidad de la información generada.
- ✓ Capacitación de usuarios, promoviendo la verificación independiente del contenido generado por LLMs y el pensamiento crítico.

LLM05:2025 Improper Output Handling (Consumo Ilimitado)

Descripción del ataque	Riesgos	Ejemplo
Ocurre cuando una aplicación basada en LLMs permite a los usuarios realizar consultas sin restricciones, lo que puede generar problemas como denegación de servicio (DoS), costos operativos excesivos, robo de modelos y degradación del servicio. Dado el alto consumo computacional de los LLMs, especialmente en entornos en la nube, los atacantes pueden explotar esta vulnerabilidad para saturar recursos, agotar presupuestos o extraer información del modelo.	Ataques de denegación de servicio (DoS) al sobrecargar el sistema con solicitudes masivas. "Denial of Wallet" (DoW), donde el atacante genera costos elevados en sistemas de pago por uso. Extracción de modelos a través de APIs, permitiendo la copia o clonación del modelo. Ejecución de consultas computacionalmente costosas, lo que puede ralentizar o colapsar el sistema. Ataques de canal lateral, donde los atacantes extraen información del modelo mediante la manipulación de entradas.	Un atacante envía una gran cantidad de solicitudes con textos de longitud variable a una API de LLM en la nube. Esto provoca un alto consumo de CPU y memoria, afectando la disponibilidad del servicio y generando costos excesivos para el proveedor.

Mitigación:

- ✓ Implementar validación de entradas, limitando el tamaño de las consultas.
- ✓ Restringir el acceso a probabilidades de salida (*logits* y *logprobs*), evitando filtraciones de información del modelo.
- ✓ Aplicar límites de tasa (rate limiting) para controlar la cantidad de solicitudes por usuario.
- ✓ Gestionar dinámicamente la asignación de recursos, evitando sobrecargas.
- ✓ Establecer tiempos de espera y limitación de procesos intensivos para evitar uso excesivo de recursos.
- ✓ Monitorear y registrar actividad anómala, detectando patrones de abuso.
- ✓ Usar watermarking[6] para rastrear el uso no autorizado de las respuestas del modelo.
- ✓ Implementar controles de acceso robustos, aplicando autenticación basada en roles (*RBAC*).
- ✓ Evitar la exposición de modelos mediante registros centralizados de ML, asegurando gobernanza y control de acceso.
- ✓ Automatizar despliegues con MLOps, aplicando restricciones en infraestructura y uso de modelos.

Conclusiones

El OWASP Top 10 para Aplicaciones LLM 2025 destaca las principales vulnerabilidades de los modelos de lenguaje grande (LLM) y su impacto en la seguridad de las aplicaciones. Entre los riesgos identificados, la inyección de prompts, el envenenamiento de datos y la filtración de información sensible representan amenazas significativas que pueden comprometer la integridad, confidencialidad y disponibilidad de los sistemas. Además, problemas como el manejo incorrecto de salida, el uso de agentes excesivos y la desinformación ponen en riesgo la confianza en estos modelos, ya que pueden generar respuestas manipuladas, incorrectas o sesgadas, afectando la toma de decisiones y la credibilidad de los sistemas basados en IA.

El abuso de los recursos computacionales, representado en el consumo ilimitado, junto con la extracción de modelos y ataques de canal lateral, plantea preocupaciones en términos de costos operativos, robo de propiedad intelectual y degradación del servicio. Asimismo, la exposición de prompts del sistema y las debilidades en vectores y embeddings facilitan ataques que pueden explotar configuraciones erróneas o controles de acceso inadecuados. Estos riesgos no solo afectan la seguridad de los LLMs, sino también su capacidad para operar de manera confiable en entornos críticos como la salud, el derecho o las finanzas.

Para mitigar estas amenazas, es fundamental implementar controles estrictos de acceso, auditoría y validación de datos, asegurando que los LLMs operen dentro de un marco seguro. Estrategias como limitación de privilegios, revisión manual en tareas críticas, uso de modelos ajustados con RAG y monitoreo constante pueden reducir significativamente el impacto de estas vulnerabilidades. La adopción de mejores prácticas en el desarrollo y despliegue de LLMs, combinada con un enfoque de seguridad proactiva y pruebas adversariales, garantizará un uso más seguro y eficiente de estas tecnologías en el futuro.

[6] Herramienta poderosa en este contexto. Se trata de un conjunto de técnicas que permiten incrustar información oculta de manera imperceptible en contenidos multimedia e incluso en texto. Esto abre un abanico de posibilidades cruciales en la lucha contra la desinformación. Recuperado de: https://blogs.uoc.edu/informatica/es/que-es-watermarking/

Canales de comunicación

El CSIRT Académico UNAD, actualmente cuenta con los siguientes canales de comunicación:

- Correo: csirt@unad.edu.co
- Página web: https://csirt.unad.edu.co

Referentes bibliográficos

Oligio. (2025). *OWASP Top 10 LLM, Updated 2025: Examples and Mitigation Strategies*. Obtenido de Oligio Security: https://www.oligo.security/academy/owasp-top-10-llm-updated-2025-examples-and-mitigation-strategies

OWASP. (2025). *Agentic AI – Threats and Mitigations*. Obtenido de OPWASP: https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

OWASP. (2025). *Agentic Threats Navigator*. Obtenido de OWASP: https://genai.owasp.org/resource/owasp-gen-ai-security-project-agentic-threats-navigator/

OWASP. (2025). *OWASP Top 10 for LLM Applications 2025*. Obtenido de OWASP: https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/