

OWASP Top 10 for LLMs, 2025

OWASP's 2025 list names the ten most critical risks in apps built on large language models. This brief covers what each one means for a team shipping RAG and agent features.

Prompt injection

Agents

RAG

● Built for image-led stories, dense insights, and motion-first reveals.

SPOTLIGHT

Visual anchor for the deck. Use this frame to establish tone, environment, and momentum.

Secondary image for product, people, or process context.

Third perspective to keep the slide cinematic and content-rich.



Made with MagicSlides

All ten risks at a glance

Grouped by where they show up in a typical LLM app.

AREA	RISK	RISK	RISK
Inputs	LLM01 Prompt Injection	LLM07 System Prompt Leakage	LLM10 Unbounded Consumption
Data	LLM02 Sensitive Info Disclosure	LLM04 Data and Model Poisoning	LLM08 Vector and Embedding Weaknesses
Outputs	LLM05 Improper Output Handling	LLM09 Misinformation	-
Build and run	LLM03 Supply Chain	LLM06 Excessive Agency	-

Prompt injection is still number one, and RAG or fine-tuning do not fully fix it.

LLMO1

Prompt injection, direct and indirect

Direct injection comes from the user's own prompt. Indirect injection hides in content the model reads, like a web page, a file or a RAG document. The text does not even need to be visible to a human.

Least privilege

Give the app its own API tokens and keep functions in code, not in the model.

Human approval

Require a person to confirm high-risk actions.

Mark untrusted content

Segregate external content and test with adversarial simulations.

OWASP scenario: a web page with hidden instructions makes the LLM insert an image link that exfiltrates the private chat.



Made with MagicSlides

Why agents need tighter limits

Excessive functionality

The agent can reach tools it does not need for the task.

Excessive permissions

Tools run with more rights than the action requires.

Excessive Agency

Excessive autonomy

High-impact actions run without a human check.

Triggers

Hallucination, prompt injection, or a compromised extension or peer agent.

The damage depends on **what the LLM can touch**, not on why it misbehaved.



Four checks before the next release

Taken from the prevention sections of LLM06, LLM07, LLM05 and LLM10.



CHECK 1

Audit tool scopes

Remove unused tools and narrow each tool's permissions.



CHECK 2

Clean the system prompt

No credentials or connection strings. Treat it as public, not as a security control.



CHECK 3

Treat output as untrusted

Validate and encode model output before it reaches a browser, shell or database.



CHECK 4

Cap consumption

Rate limits and input size limits guard against denial of wallet.

Run the checklist on every feature that adds a tool or a data source.

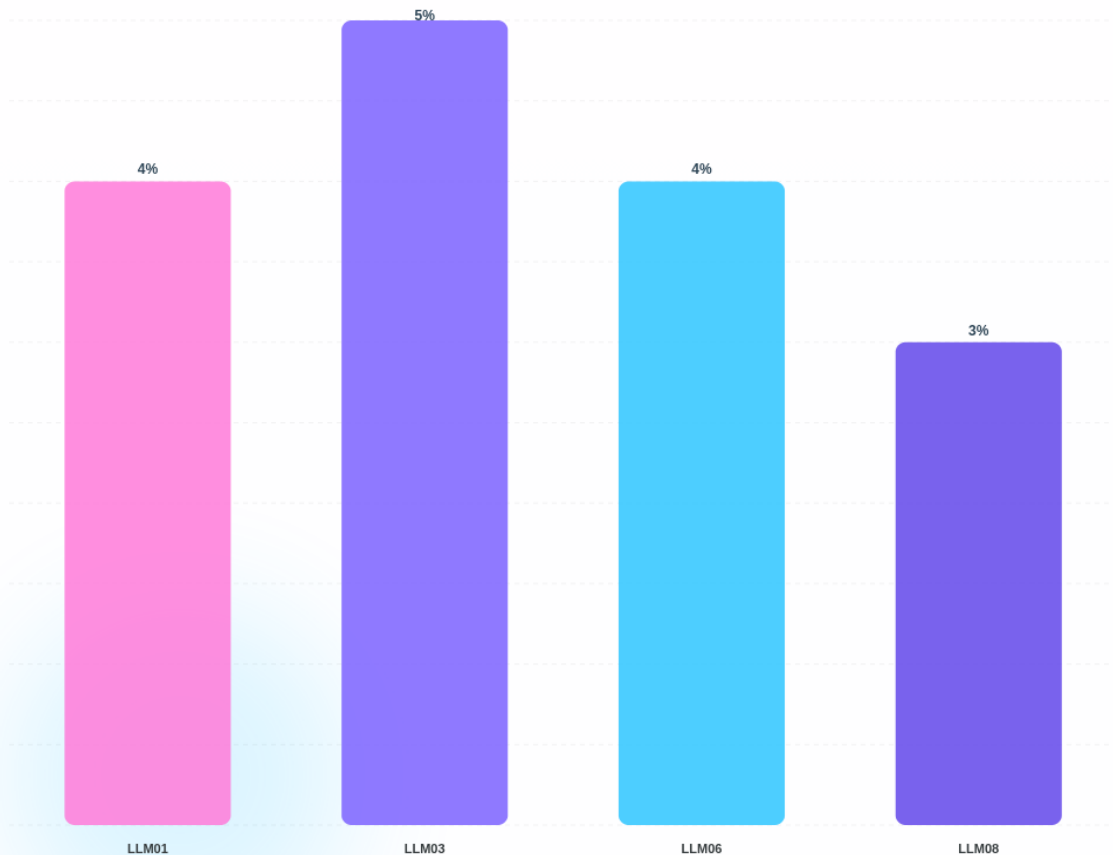
PRE-RELEASE

Made with MagicSlides

Where the guidance goes deepest

Pages per risk, selected

45-PAGE PDF



Pages per entry in the OWASP PDF, a rough guide to detail.

Core Analysis

Supply chain

Gets the longest entry, covering models, LoRA adapters and third-party packages.

Agents

Excessive agency gets as much space as prompt injection.

RAG

Vector and embedding weaknesses is a new entry focused on RAG.

