LLM02:2026

Sensitive Information Disclosure

OWASP LLM Top 10 · 2026 rewrite

Why It Matters

#2

Holds steady in the Aug 2026 OWASP rewrite — unchanged rank alongside LLM01

Top 2

Cited with LLM08 Hidden Context Exposure as the most common production vulnerabilities after prompt injection

RAG-backed

Most enterprise assistants are RAG-backed — RAG leakage + context pull are the dominant real-world attack shapes

Three Distinct Mechanisms

Mechanism 1: Training-Data Extraction

Duplication Count Drives Memorization

Unique passage

Single gradient nudge — overwritten by unrelated training signal. Not memorized.

10,000x-duplicated boilerplate

Repeated updates compound in the same direction. Verbatim memorization becomes the loss-minimizing move.

Not model capacity or "outlier-ness" — duplication count is the dominant factor.

Extraction Technique: Prefix Attacks

Known Prefix → Verify the Completion

Unlike a direct question, a known prefix's completion can be checked word-for-word against a known suffix — this is what makes extraction verifiable.

Verification

Known prefix, known completion. Proves memorization exists — research / audit framing.

Genuine Extraction

Known prefix, UNKNOWN completion — real sensitive data (e.g. name-known / SSN-unknown template records). The actual production risk.

Mitigation: Deduplication

Fix It at the Root Cause

Fuzzy / near-duplicate deduplication

Primary mitigation — removes the duplication signal at the training-data stage, not just exact-match copies

PII masking / obfuscation

Complementary defense-in-depth layer, applied on top of deduplication

Mechanism 2: RAG / Context-Window Leakage

A Different Mechanism Entirely

Not weight-based memorization — content injection. Unauthorized documents enter the context window at retrieval time.

Risk enters here: an unauthorized document can be retrieved and injected into context

Mitigation: Retrieval-Time Access Control

It's an Authorization Problem

Wrong fix: Obfuscation

Defeats RAG's actual purpose — surfacing real values (e.g. a user's own account balance) back to the right person.

Right fix: Access Control

Never let unauthorized documents enter the context window at all — retrieval-time authorization.

Same principle as SQL row-level access control.

"Habit, Not Guarantee"

Prompting the model to behave has no mechanical enforcement inside a forward pass.

LLM01

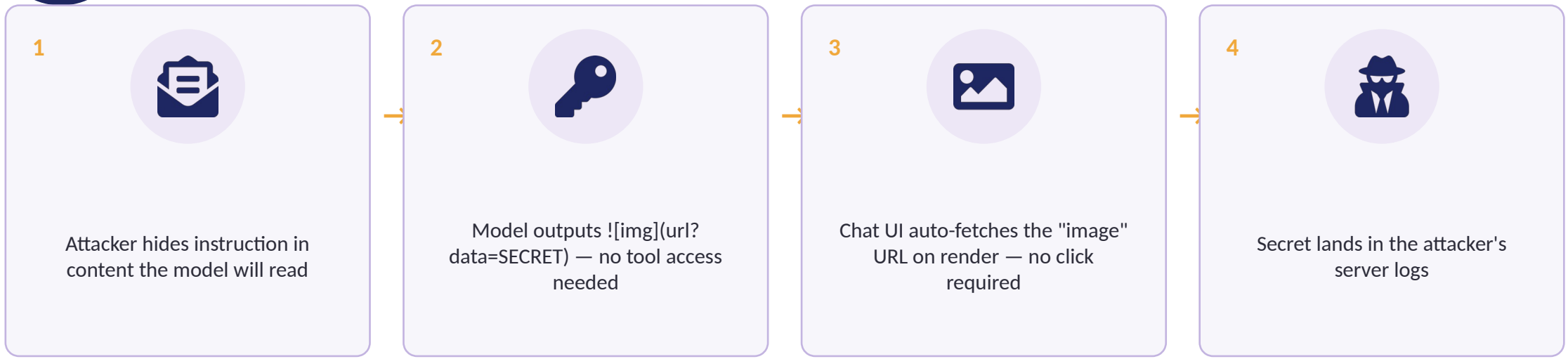
Privilege-limits mitigation

LLM02

Retrieval-time access control

Mechanism 3: Data Exfiltration Channels

Zero-Click, Via Markdown Rendering



Reference case: EchoLeak (CVE-2025-32711, CVSS 9.3) — Microsoft Copilot, fully zero-click

Mitigation: URL Allow-Listing

Closed Space Beats Open Space

Data-Pattern Filtering

Open, unbounded space of "sensitive" — SSNs, addresses, split values, context-dependent secrets. Weak, complementary layer only.

URL Allow-Listing

Closed, enumerable set of trusted domains. Binary membership check — the primary, structural mitigation.

Same closed-vs-open principle as privilege limits and schema-constrained output (LLM01).

Mitigation: Egress Filtering

A Second, Independent Layer

App-Level Allow-List

Checked inside the renderer's own code — vulnerable to bugs or a different code path that forgets the check.

Network-Level Egress Filter

Independent system, closer to the OS/hardware boundary — blocks the connection regardless of app-code bugs or intent.

Same defense-in-depth principle as gVisor / Firecracker: one narrow, checked gate — enforced redundantly.

Full Mitigation Map

Mechanism	Root Cause	Primary Mitigation
Training-data extraction	Duplication compounds gradient updates	Deduplication + PII masking
RAG / context-window leakage	Unauthorized content enters context	Retrieval-time access control
Data exfiltration	Sensitive data smuggled via auto-fetched URLs	URL allow-listing + egress filtering

LLM02:2026 — Complete

"Sensitive information disclosure isn't one bug — it's three separate structural gaps: what gets memorized, what gets retrieved, and what gets rendered. Each needs its own outside-the-model fix."