AI Model Risk Management: NIST AI RMF and LLM Red-Teaming

Erik Jones April 9, 2026

Executive Summary

Generative AI moved from research project to production system faster than any technology cohort in the past two decades, and most organizations now run LLM-powered features in production without an AI-specific risk framework wrapped around them. This guide explains the regulatory landscape, the NIST AI Risk Management Framework's operational shape, the OWASP Top 10 for LLMs, and the toolchain that turns AI risk management from a paper exercise into an operational program.

Why AI Model Risk Management Matters

The problem space is not theoretical: **prompt injection** attacks have leaked customer data, **jailbreaks** have produced regulated outputs

(medical advice, legal recommendations) the system was supposed to refuse, and **hallucinated outputs** have been cited in court filings with consequences for both the model deployer and the downstream user.

The regulatory environment has caught up. The **EU AI Act** entered force in August 2024, with high-risk AI obligations phased in through 2026. The **NIST AI Risk Management Framework** provides the operational counterpart for organizations not subject to the EU regime. **ISO/IEC 42001 (Artificial Intelligence Management System)**, published in December 2023, is becoming the certification of choice for organizations needing a third-party-attested AI governance program. SOC 2 reports increasingly include AI-specific control sections. The compliance pressure is real and growing.

NIST's framework guidance puts it directly: *"AI risk management is a journey, not a destination, and the framework is intended to be operationalized as a continuous practice — not a one-time assessment."*

NIST AI Risk Management Framework

The NIST AI RMF (1.0, January 2023, with the **Generative AI Profile** published July 2024) organizes AI risk management around four functions.

The Four Functions

Govern: Cultivate a culture of risk management. Includes policies, accountability structures, training, and integration with enterprise risk management. The function most often skipped at startup scale and most often cited in regulatory findings.

Map: Establish context. AI system inventory, intended use cases, deployment context, stakeholder identification, impact assessment. Without a map, you cannot reason about risk.

Measure: Analyze and benchmark. Performance metrics, accuracy testing, bias and fairness analysis, robustness evaluation, security testing, explainability documentation.

Manage: Prioritize and act on risks. Risk treatment decisions, monitoring, incident response specifically for AI, model lifecycle management.

The Cycle, Not the Sequence

The four functions are not sequential — they cycle. New deployments trigger Map; production drift triggers Measure; identified risks trigger Manage; the entire cycle is overseen by Govern. Treating the framework as a one-pass checklist misses the operational point.

The Generative AI Profile

The July 2024 profile addresses LLM-specific concerns: confabulation, dangerous content generation, intellectual property, obscene content, information integrity, information security, value chain risks. The

profile is the operational guidance most relevant to SaaS companies deploying LLM features.

OWASP Top 10 for LLMs (2025)

The OWASP Top 10 for LLM Applications defines the dominant threat model for LLM-powered systems. The categories matter because each maps to specific defensive controls.

The Categories

LLM01 Prompt Injection: direct (user input manipulates the model) and indirect (untrusted content like emails or web pages embeds malicious instructions). The most-exploited LLM vulnerability class in production.

LLM02 Sensitive Information Disclosure: models leak training data, system prompts, or context window contents. Mitigated through output filtering and context isolation.

LLM03 Supply Chain: compromised model weights, poisoned training data, malicious dependencies in the AI stack.

LLM04 Data and Model Poisoning: training-time and fine-tuning-time attacks that bias model behavior.

LLM05 Improper Output Handling: treating LLM output as trusted input downstream — classic injection patterns return when LLM output flows into SQL queries, shell commands, or rendered HTML.

LLM06 Excessive Agency: AI agents granted permissions beyond what the use case requires. Risk surface expands rapidly as agentic capabilities grow.

LLM07 System Prompt Leakage: system prompts extracted by adversarial users, exposing business logic and security boundaries.

LLM08 Vector and Embedding Weaknesses: attacks against RAG systems, vector store poisoning, embedding inversion.

LLM09 Misinformation: hallucinated outputs presented as authoritative.

LLM10 Unbounded Consumption: resource exhaustion through unbounded generation, prompt amplification, or wallet attacks against pay-per-token systems.

Practical Implementation

The AI Inventory

Catalog every AI system, including third-party APIs (**OpenAI, Anthropic, Cohere**) and embedded ML features. Capture: use case, data classifications processed, model provider, deployment environment, owner, last review date. The inventory drives every other control.

Threat Modeling

Apply **STRIDE** or the OWASP LLM Top 10 to each system. Document threat actors, attack vectors, and current mitigations. The threat model is itself a Map artifact for NIST AI RMF.

The Red-Team Toolchain

Garak (NVIDIA, open-source LLM vulnerability scanner) — run baseline scans pre-deployment and quarterly thereafter

PyRIT (Microsoft, automated red-teaming framework) — programmatic adversarial testing at scale

Promptfoo (test framework with adversarial templates) — CI-integrated regression testing for LLM behavior

Lakera Red (commercial) — managed red-team platform with continuously-updated attack catalogs

Runtime Guardrails

LLM Guard — open-source input/output filter

Lakera Guard — commercial prompt-injection and PII filtering

NeMo Guardrails (NVIDIA) — programmable conversational rails

Self-built filters — for narrow use cases where commercial guardrails are overkill

Apply guardrails at both input (sanitize prompts) and output (filter responses) layers. Log everything for incident analysis.

Evaluation Suite

Define accuracy benchmarks per use case. For generative outputs, define **refusal-rate benchmarks** (the system should refuse certain categories of requests) and **consistency benchmarks** (similar inputs should produce semantically similar outputs). Run the suite continuously in CI.

Incident Response

AI-specific runbooks. What happens when prompt injection is detected in production? When a hallucinated output reaches a customer? When a regulator asks for a model card? The runbook is the difference between a contained incident and a public one.

Common Pitfalls

Treating AI like traditional software: traditional software is deterministic; AI systems are probabilistic, drift over time, and behave differently with adversarial inputs than with benign ones. Test methodologies built for deterministic systems miss the entire class of AI-specific failures.

Conflating safety and security: safety controls (refuse harmful content, avoid bias) protect users from the model. Security controls (prevent prompt injection, protect system prompts, isolate context) protect the system from users. Both are required; neither substitutes for the other.

No AI inventory: you cannot govern what you have not catalogued.

No continuous testing: a model that passed red-team testing in March drifts by September. Continuous evaluation in CI catches the drift.

Over-broad agentic permissions: the model can read everything, so prompt injection becomes data exfiltration.

Vendor-only assurance: "we use OpenAI so we are secure" misunderstands the threat model. The integration boundary, system prompts, and downstream output handling are all the deployer's responsibility.

Measured Outcomes

AI inventory complete within 30 days, refreshed quarterly

NIST AI RMF mapped across Govern, Map, Measure, Manage with named owners per function

OWASP LLM Top 10 threat model documented per AI system

Red-team baseline via Garak and PyRIT pre-deployment, refreshed quarterly

Runtime guardrails at input and output, integrated with logging and incident response

Continuous evaluation in CI catching behavior drift

Audit-ready model cards and risk documentation satisfying SOC 2, ISO 42001, and EU AI Act expectations

How Jacobian Helps

Our team works at the intersection of AI engineering and security audit — we build production AI systems and we test them. That combination is rare and matters operationally: we understand what the model is supposed to do, where the integration boundaries leak, and what attackers will probe first. We map your AI systems against NIST AI RMF, run red-team campaigns using Garak and PyRIT, design runtime guardrails that fit your latency and cost constraints, and produce model cards and risk documentation that satisfy SOC 2, ISO 42001, and EU AI Act expectations. When you are ready to bring AI governance into your existing compliance program, we crosswalk it; when you need to stand up a program from scratch, we build it.

Resource Details

Author: Erik Jones

Published: April 9, 2026

Categories:

AI Security | Risk Management | Compliance

Download

Full Document

About This Resource

A practical guide to AI model risk management for SaaS companies deploying LLMs and generative AI. Covers NIST AI RMF, OWASP Top 10 for LLMs, red-teaming methodology, and the operational controls that distinguish responsible deployments.

Categories:

AI Security | Risk Management | Compliance