# An LLM Security Guide for CIOs

paloalto NETWORKS | PRISMA AIRS™

In the heat of the AI race, we've seen enterprises deploy AI applications without much thought to hardening security systems. **Your role as CIO is to move LLM security from a theoretical concern to a proactive, integrated part of your cybersecurity program**.

The solution is not just technical; it's operational. By establishing a clear LLM security playbook, you can ensure that your organization's AI initiatives are not only innovative but also secure and resilient.

# Welcome to the New Frontier

It's rare that a new technology redraws the boundaries of the way we work overnight — but that's exactly what happened when large language models (LLMs) broke into the mainstream. LLMs have moved from labs into daily operations, creating a new attack vector.

AI is moving beyond the experimental phase, as it becomes a core part of business operations. In fact, 71% of organizations are regularly using generative AI in at least one business function. This rapid integration means that many companies are fine-tuning models to fit their specific needs, often treating security as an afterthought.

**This guide will provide a strategic framework to help secure your AI journey**. We will explore how LLMs work and the unique security risks that come with this new technology.

# Understanding the LLM: A Powerful Pattern-Matcher

Think of LLMs as advanced pattern-matchers: when you hear 'Roses are red, violets are…' your brain fills in 'blue' — and the model does the same. In the same manner, the language model, having seen this pattern countless times in its training data, makes that exact same prediction.

**This process involves a few key stages, which you'll hear about often when discussing LLMs:**

- **Training**: The model is pre-trained on a massive dataset, learning grammar, context, and general knowledge—like a student who has read every book in a library.
- **Fine-tuning**: It's then refined on a smaller, specialized dataset for a specific task, similar to graduate-level study.
- **Prompting**: Finally, users interact by giving commands or questions, and the model generates a response based on its training.

# The New Threat Landscape: When AI Is a Target

## Every AI journey starts with a model.

For most organizations, training one from scratch is prohibitively expensive and time-consuming. Instead, they start with a pre-trained foundation model and fine-tune it with proprietary data—FAQs, policies, or brand guidelines—to fit their business needs. This approach delivers customization and speed without the massive cost of building a model from the ground up. Once fine-tuned, the model is deployed through APIs to power the applications employees and customers use every day.

As shown in the diagram above, there are critical threats you may encounter along the way of developing your LLM, as defined by the **OWASP Top 10 for LLM Applications**:

- **Tricking the AI with commands**: Prompt injection is like social engineering for LLMs. Attackers use crafted prompts to bypass safeguards and trick models into revealing sensitive data or generating harmful content.
- **Corrupting the AI brain**: Through data and model and poisoning, attackers inject malicious or fake data into training sources. This can bias the model, spread misinformation, or create hidden backdoors.
- **Letting the AI's response cause damage**: If an app uses an AI's output without checking it, attackers can insert malicious code that compromises systems and user accounts.
- **Leaking sensitive data**: LLMs can expose private information if they were trained on sensitive data or if user inputs are reflected in other conversations—especially when connected to internal company systems.

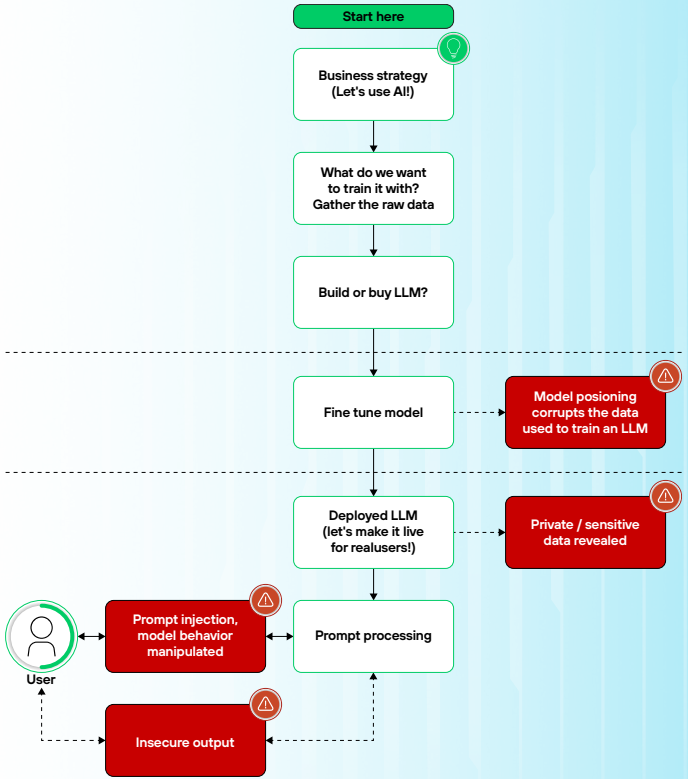


**Figure 1:** The AI development flow as defined by the OWASP Top 10 for LLMs, including potential threats that can occur at different stages

# Operationalizing AI Security

LLMs can reveal sensitive data in unexpected ways, especially when tied to internal systems—making them both a business enabler and a potential liability. For CIOs, that means AI security can't live in the margins of your program. It has to be operationalized: embedded into day-to-day processes, linked with your broader security stack, and continuously adapted as threats evolve.

This is not about one-off fixes or bolt-on safeguards. It's about creating a repeatable framework that gives your team visibility, enforces guardrails, and ensures AI adoption doesn't outpace security. The following five practices form the foundation of that framework—steps you can take today to reduce risk, strengthen defenses, and keep innovation moving safely forward.

### 1. Get ahead of shadow AI

The shadow AI problem has grown far beyond a few employees using ChatGPT. Across industries, workers are building unsanctioned apps and agents. They can bypass governance, expose intellectual property, and introduce unvetted code, all while operating outside IT's line of sight.

Prisma AIRS Run Time Security addresses this directly with built-in discovery. By analyzing network traffic and intercepting API calls, it can surface where unapproved LLMs and agents are in use. Runtime then inspects prompts and responses in real time, flagging sensitive data exposure, malicious content, and unsafe outputs. Every event is logged with rich context—who used what model, when, and how—and forwarded to your SIEM for unified visibility.

Discovery is only the first step. With Runtime in place, organizations can enforce policy controls to block risky behaviors, while offering secure, sanctioned alternatives for employees. This turns shadow AI from a hidden liability into a managed, monitored, and ultimately productive part of the enterprise.

### 2. Uplevel your team's skills

With shadow AI in check, the next challenge is building deeper security expertise around AI itself. Traditional skills in network and endpoint defense don't fully prepare teams for model-level threats, where risks like prompt injection, model manipulation, and data leakage demand new ways of thinking. Teams need training to recognize and respond to risks like prompt injection and data leakage in production to scale that knowledge, many organizations are formalizing an **AI Security Center of Excellence**. A CoE not only governs risk but also runs continuous red teaming and sets guardrails for secure adoption across business units. Acting as both an enforcer and a strategic advisor, it embeds security into the design of every new AI initiative.

But internal investment alone isn't enough. The threat landscape moves faster than most enterprises can track, making **external partnerships with specialized vendors** and consultancies essential. These partners bring frontline experience and real-time intelligence, helping teams balance agility with discipline as they secure their AI programs. Resources like our **Cyberpedia** can raise baseline awareness across the enterprise, but the goal is clear: build a security function that treats AI as a permanent capability, not a passing experiment.

### 3. Integrate and automate

With interconnected systems and cloud adoption, the old approach no longer flies. AI-specific threats now need to be visible and actionable inside your broader security stack.

Consistency is key here. Every AI system event: prompt details, failed injections, sensitive data exposure, API usage — must be logged in a standardized format. Logs provide context, metrics reveal trends, and together they integrate cleanly into SIEM and GRC platforms for a unified view of risk.

From there, automation links AI activity with broader enterprise signals, enabling faster, policy-aligned responses. Prisma AIRS Runtime streamlines this process by scanning prompts and responses in real time, detecting threats like prompt injections and data leaks.
Its API and Network Intercept generate rich, standardized logs that flow into Strata Cloud Manager or your SIEM, transforming raw AI activity into actionable security intelligence.

### 4. Red team for continuous validation

The LLM threat landscape shifts daily, what's secure now may be vulnerable tomorrow. Security can't be treated as a one-time project — it needs to be a constant cycle of testing and adaptation. Red teaming provides one of the most effective ways to validate defenses against real-world threats.

Automated red teaming agents continuously probe your systems with attacks such as prompt injection, data exfiltration, jailbreaking, and output manipulation. These exercises expose how guardrails can be bypassed and where sensitive data may leak, giving your teams the insights to harden defenses before adversaries strike.

The value multiplies when findings are fed directly into development and security workflows. Instead of living in isolation, they drive quick remediation and make security part of every AI deployment from the start. Prisma AIRS takes this further with its adaptive Red Teaming Agent, which replicates attacker behavior at scale. The solution delivers detailed remediation guidance, and integrates results with Runtime Security — turning adversarial testing into actionable policies that evolve as fast as the threat landscape.

### 5. Implement runtime security

Think of runtime security as the last line of defense, a security guard on duty 24/7. Unlike static checks that only validate code once, runtime protection continuously inspects prompts, responses, and system activity as they happen. This real-time visibility is what makes it possible to stop malicious behavior before it can escalate into data loss or model compromise.

As users interact with your AI applications, a runtime layer monitors for threats like prompt injections, data exfiltration, or anomalous behavior that signals misuse. It doesn't just detect risks — it actively blocks them, ensuring unsafe outputs never reach business systems or sensitive data.

Prisma AIRS Runtime delivers this capability through API and Network Intercept, capturing activity in motion with full context on who accessed what, when, and how. Every event is standardized into logs and metrics, then surfaced in Strata Cloud Manager or forwarded to your SIEM for correlation with the rest of your enterprise security telemetry. The result is a proactive, always-on shield that closes visibility gaps, strengthens defenses, and allows innovation to move forward without compromise.

# Building with a Secure AI by Design Mindset

But runtime defense alone isn't enough. To truly future-proof your AI strategy, security needs to be built in from the ground up. That means designing applications with resilience in mind — not just reacting to threats after they appear.

Successfully and securely integrating LLMs into your organization requires a proactive mindset that extends across the entire AI lifecycle.

## Giving LLMs a private knowledge base with RAG

A powerful alternative to expensive and time-consuming model fine-tuning is Retrieval-Augmented Generation (RAG). **Let's bring RAG into the picture:**

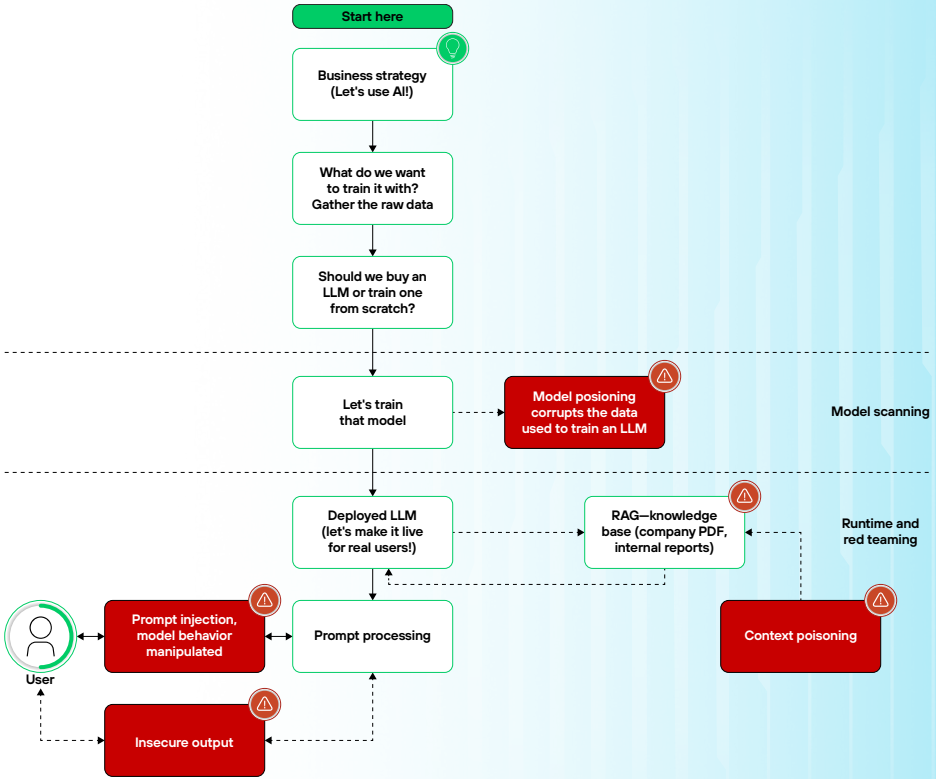


**Figure 2:** The AI development flow, using RAG to supplement the training data

### How It works

Instead of fine tuning and modifying the model's brain, you connect it to a private, verifiable knowledge base (like a database of your company's PDFs, internal reports, or a live news feed).

When a user asks a question, the RAG system first finds the most relevant information in your documents. It then takes that information and packages it up as a "context" for the LLM to use. The LLM then answers the question using its general knowledge plus the new, specific context you provided.

### Why Use RAG

RAG addresses three of the biggest challenges with LLMs: hallucinations, outdated knowledge, and data security. Hallucinations happen when models generate confident but incorrect answers because they're relying on incomplete or biased training data.

By grounding responses in trusted, verifiable sources, RAG significantly reduces errors. Unlike traditional LLMs, which are locked to a fixed training dataset that quickly goes stale and is costly to retrain, RAG pulls in fresh, domain-specific information at query time. **The result: outputs that stay accurate and relevant without the need for constant retraining.**

Finally, RAG protects sensitive data by keeping it in your own databases instead of embedding it into the model. Information is only retrieved when needed, and with proper access controls in place, you can prevent the risks of memorization or unintended data leakage.

### The Next Evolution: LLM Agents

While a basic LLM is a powerful prediction engine, an LLM agent is a distinct architectural pattern that can autonomously complete complex tasks. Unlike a traditional LLM that receives an input and produces a static output, an agent goes through a continuous process of observing, thinking, and acting to achieve a goal.

LLM agents operate on an iterative Observe-Think-Act loop, a structured process that enables it to solve problems autonomously:

- **Observe**: The agent receives a high-level goal or an external event. It assesses the current situation by "observing" the results of its previous actions and gathering any necessary information from its memory or tools.
- **Think**: Using its core language model, the agent performs advanced reasoning. It breaks down the user's goal into smaller, manageable sub-tasks, formulates a step-by-step plan, and continually reflects on its progress, adapting the plan as needed.
- **Act**: Based on its reasoning, the agent decides which external tool to use and what specific action to perform. This dynamic interaction with tools allows agents to move beyond static knowledge and directly perform tasks that require real-time information or interaction with enterprise systems.

This ability to act on its own, based on complex reasoning, is what transforms the LLM from a passive output generator into an active system that is responsible for autonomously interacting with the real world.

As foreshadowed in every movie where man meets machine, there are serious risks.
**Let's start with the more inherent weak points of LLMs.**

# The New Class of Agentic Threats

While hallucinations are a known LLM risk, agent autonomy introduces a new and more dangerous class of threats. Unlike a passive LLM, an agent's ability to act on a goal—connecting to databases, communicating with the outside world, or executing code—turns a theoretical vulnerability into a real-world risk multiplier.

As you can see in the diagram above, there are dynamic ways attackers can interfere with how an LLM agent works.

- **Tools**: When the agent calls external tools (e.g., payment processors, calendars, CRM systems), it inherits supply chain risks such as poisoned open source models or libraries — if even one of those outside services is compromised, the attacker has a direct path into the agent.
- **Permissions and trust**: Agents are often given broad access so they can work efficiently, but that also means a single compromise can unlock a lot of power, letting attackers act at machine speed before anyone notices.
- **Memory**: Memory provides the agent with past context, but that also opens the door to memory poisoning, where bad data gets stored and later reused, leading to leaks or flawed reasoning.
- **Planning**: Attackers can mess with planning (the part of the system that sequences steps) by hijacking goals and redirecting the agent toward harmful tasks.

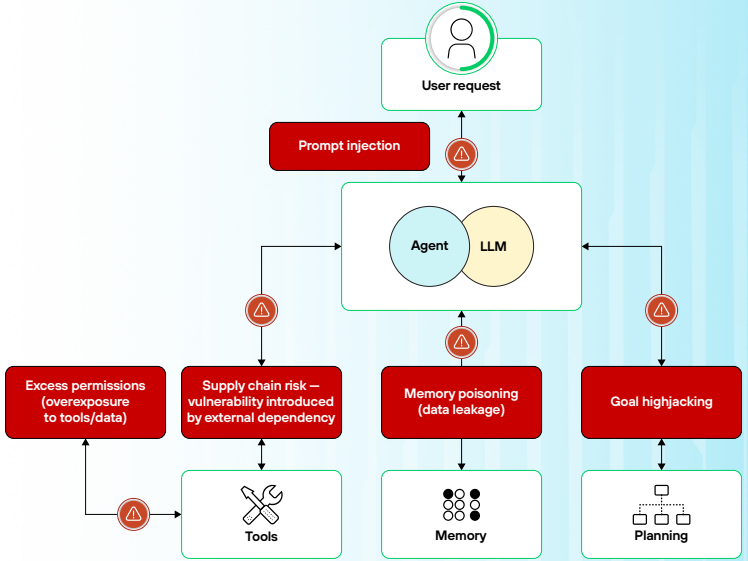


**Figure 3:** A user/agent interaction showing where certain vulnerabilities could be exploited

Agentic threats can creep in at every stage of how an LLM agent operates. External tools may carry supply chain risks, broad permissions amplify the impact of a single compromise, memory can be poisoned with false data, and planning logic can be hijacked to redirect tasks. Combined, these weaknesses make agent autonomy a powerful, and potentially dangerous, attack vector.

### Buy vs. Build: How to Select an AI Security Vendor

Securing LLMs and AI agents is not a problem you can afford to defer. The cost of inaction is steep: data leaks, regulatory penalties, reputational damage, and stalled innovation quickly outweigh the investment in a security platform. Building an in-house solution may look appealing at first, but the reality is a long, expensive, and resource-heavy journey — one few organizations can sustain at the speed attackers move.

- **Choose a platform with breadth**: In-house builds often result in silos, blind spots, and mounting maintenance overhead. A unified platform eliminates that complexity, providing a single pane of glass across cloud, network, identity, and data. This cross-layer visibility lets you follow threats end-to-end and enforce consistent controls across the AI lifecycle.
- **Prioritize proactive defense**: Custom-built solutions typically lag behind the threat curve, leaving teams reactive. Leading platforms embed AI red teaming and runtime protection that continuously stress-test your environment, exposing vulnerabilities like prompt injection or insecure outputs before adversaries exploit them.

- **Focus on seamless integration**: DIY approaches often require stitching together tools and APIs, creating fragile connections that strain security teams. A purpose-built platform integrates natively with enterprise systems, normalizing logs, aligning with SIEM/GRC workflows, and enabling fast, automated responses without adding operational burden.
- **Future-proof your investment**: AI threats evolve daily, and keeping pace requires continuous intelligence, tuning, and expertise. With the right vendor, you get not just technology but a partner who absorbs the cost of ongoing innovation and lets your team focus on securely scaling AI.

**The choice is simple:**
**Spend years building piecemeal defenses, or accelerate today with a platform designed to secure the full AI lifecycle.**

paloalto NETWORKS | PRISMA AIRS™

## About Palo Alto Networks

Palo Alto Networks is actively innovating in AI Security.
Learn more about Prisma AIRS, the world's most comprehensive AI security solution.

## Ready to learn more? Contact us.