Mend.io

AI Security Governance: A Practical Framework for Security and Development Teams

How to build durable AI governance that keeps pace with how your teams actually work — without turning every AI adoption decision into a security standoff

The governance problem nobody planned for

AI adoption inside organizations rarely starts with a strategy. It starts with a developer using Copilot to ship faster, a data analyst querying a new LLM tool for reporting, or a product team embedding a third-party model to accelerate a feature. By the time security hears about it, the AI is already in production.

That gap — between how fast AI enters the organization and how slowly governance catches up — is where risk lives. And most organizations aren't equipped to close it.

This guide doesn't assume you have everything figured out. It's built for security teams that are trying to get ahead of AI sprawl, and for development teams that want to build with AI without creating problems downstream. The goal is a governance model that's realistic to implement, useful across teams, and durable enough to evolve as AI capabilities change.

What this guide covers:

- How to build an AI asset inventory that actually stays current
- A risk classification approach you can apply at scale
- Access controls and data governance principles specific to AI systems
- Supply chain security for models, datasets, and AI components
- Monitoring, behavioral analysis, and incident response for AI workloads
- How to build policy that teams will follow — and tooling that enforces it

WHO THIS GUIDE IS FOR

Security architects, AppSec leads, and CISOs building or maturing an AI governance program. Also useful for engineering managers and platform teams evaluating how to integrate AI responsibly without adding friction to development.

Building an AI asset inventory

You cannot govern what you cannot see. The first step in any AI security program is knowing what AI is already operating in your environment — and that number is almost always higher than teams expect.

AI assets span a wider surface area than traditional software components. They include:

- AI-assisted development tools embedded in IDEs (Copilot, Cursor, Codeium, etc.)
- Third-party AI APIs consumed by applications (OpenAI, Anthropic, Google Gemini, etc.)
- Open-source models pulled from Hugging Face, GitHub, or similar registries
- AI components bundled inside SaaS tools your teams use (Salesforce Einstein, Notion AI, Slack AI, etc.)
- Internal models trained or fine-tuned on your own data
- AI agents and automation pipelines with autonomous decision-making capability

What your inventory needs to capture

A spreadsheet is a reasonable starting point. It won't stay current, but it creates a baseline. Mature organizations move to a registry that's populated through automated discovery and validated continuously. Regardless of the format, each AI asset entry should capture at minimum:

- Asset name and version (model name, tool name, API endpoint)
- Purpose and use case (what it does, what decisions it influences)
- Business owner and technical owner
- Data inputs: what data does this AI process, and at what sensitivity level?
- Data outputs: what does the model return, and where does that output go?
- External dependencies: is the model hosted externally, or on-premises?
- Integration points: which applications, pipelines, or services consume this AI?
- Access scope: which teams and systems have access?

Keeping the inventory current

The biggest failure mode for AI inventories is staleness. Developers add new tools quickly; the registry falls behind. A few practices that help:

- Integrate AI asset declaration into your standard onboarding process for new tools and services
- Use network egress monitoring to detect calls to known AI API endpoints
- Require AI disclosure in architecture review and threat modeling templates
- Run periodic audits against your software bill of materials (SBOM) for AI-specific components

GOVERNANCE NOTE

Shadow AI, AI tools in use that security hasn't approved or catalogued, is the most common inventory gap. Treat the discovery of shadow AI as a non-punitive process. If developers feel they'll get in trouble for disclosing, they won't disclose.

An AI risk classification framework

Not all AI deployments carry the same risk. A Copilot integration helping engineers write boilerplate code is a different threat surface than an AI agent with write access to a production database. Treating everything as equally high risk leads to security theater — and treating nothing as high risk leads to incidents.

A practical classification approach scores AI assets against a small set of criteria, producing a risk tier that determines the depth of review, the controls required, and the cadence of monitoring.

IMPORTANT

Risk scores should be reassigned whenever an AI asset's access scope, data inputs, or decision authority changes. A model that starts as Tier 1 can become Tier 3 after a single integration change.

Risk scoring dimensions

Score each AI asset across these dimensions on a 1–3 scale (Low / Medium / High):

Data Sensitivity

1:	2:	3:
Processes only public or non-sensitive data	Processes internal business data	Processes PII, financial data, IP, or regulated data

Decision Authority

1:	2:	3:
Advisory output only; humans make final decisions	Influences significant decisions but with human review	Autonomous or semi-autonomous decision-making with limited human oversight

System Access

1:	2:	3:
Read-only or no direct system access	Read-write access to non-critical systems	Write access to critical systems, infrastructure, or customer-facing services

External Exposure

1:	2:	3:
Internal use only, isolated environment	Internal with potential for indirect external impact	Customer-facing, public API, or externally accessible

Supply Chain Origin

1:	2:	3:
Reputable vendor with published security program and SLA	Open-source with active maintenance and known provenance	Unknown origin, unsupported, or community models without vetting

Interpreting the score

Score 5–7: Tier 1 (Low Risk) — Standard security review; lightweight monitoring; no special controls required

Score 8–11: Tier 2 (Medium Risk) — Enhanced review; access controls required; quarterly behavioral audits

Score 12–15: Tier 3 (High Risk) — Full security assessment; design review required; continuous monitoring; IR playbook required before deployment

Limiting what AI can see and do

The most common AI security failures aren't model vulnerabilities — they're access control failures. An AI with access to more data than it needs, or more system permissions than its function requires, dramatically expands your blast radius when something goes wrong.

Principle of least privilege for AI

Apply the same least-privilege principles to AI systems that you apply to human users and service accounts:

- Define the minimum data inputs necessary for each AI function — and enforce those boundaries
- Use read-only access wherever write access isn't explicitly required
- Scope API keys and tokens to specific resources, not broad service-level access
- Avoid shared credentials between AI systems and human users
- Rotate credentials used by AI systems on a defined schedule

Data classification and handling rules

Before any AI system processes data, define the rules for how different data classifications can flow into AI models. Consider these boundaries as a starting point:

- Regulated data (PII, PHI, PCI) should require explicit approval before use as AI training data or model input
- Proprietary IP should be prohibited from training external commercial models unless you have a contractual data protection agreement
- Customer data should be subject to your existing privacy policy — if your privacy policy doesn't address AI processing, update it before deployment
- Logs, telemetry, and internal operational data should be reviewed for embedded sensitive values before feeding into AI analytics or observability tools

Output controls

Data governance for AI isn't only about what goes in. AI outputs can themselves become a data leak vector — especially when models generate content that reconstructs or infers sensitive information. Governance controls for AI outputs include:

- Output filtering for regulated data patterns (SSNs, credit card numbers, API keys)
- Logging AI-generated outputs in applications where those outputs influence business decisions
- Preventing AI output from being routed directly to external users without human review for high-risk use cases
- Treating AI-generated code as untrusted input — subject to the same SAST, SCA, and secrets scanning as human-written code

Securing the models and components you didn't build

AI systems are supply chain systems. When you deploy a model, you're also inheriting the security posture of whoever trained it, whatever dataset it learned from, and whatever dependencies were bundled with it. That supply chain is long, often opaque, and not well-governed by the broader ecosystem.

Traditional SCA tools cover open-source software dependencies. AI supply chain security extends that concept to model artifacts, datasets, fine-tuning inputs, and inference infrastructure. The attack surface is different; the governance principles are similar.

Model vetting before deployment

Third-party models — whether pulled from public registries or purchased from vendors — should go through a vetting process before production deployment. At minimum:

- Verify model provenance: who trained it, on what data, and where is that documented?
- Scan model files and associated dependencies for known vulnerabilities (particularly for models distributed as Python packages with transitive dependencies)
- Review the model card or documentation for disclosed limitations, known failure modes, and training data sources
- Conduct red teaming or adversarial probing for models that will process untrusted user input
- Evaluate the model's behavior on edge cases relevant to your use case before deployment

Training data integrity

If your organization trains or fine-tunes models, the integrity of your training data becomes a security concern. Data poisoning — where malicious data is injected into training pipelines to manipulate model behavior — is an active threat, particularly for organizations using data from uncontrolled public sources.

- Establish provenance tracking for training datasets: where did this data come from, and has it been modified?
- Apply anomaly detection to training datasets before ingestion
- Maintain versioned, immutable copies of training data used for each model version
- Document what data was used to train each model version — this is the foundation of an AI SBOM

AI bill of materials (AI-BOM)

Just as a traditional SBOM documents the open-source components in your software, an AI-BOM documents the components of your AI systems. A complete AI-BOM includes:

- Model name, version, and source
- Training data references and versions
- Fine-tuning datasets (if applicable)
- All software dependencies required to run the model
- Inference infrastructure components
- Known vulnerabilities and their remediation status

REGULATORY CONTEXT

Several emerging AI regulations and frameworks — including the EU AI Act and NIST AI RMF — explicitly reference supply chain transparency requirements. An AI-BOM positions you well for compliance regardless of which framework your organization aligns to.

Detecting and responding to AI-specific threats

Most security monitoring infrastructure wasn't built with AI workloads in mind. Traditional SIEM rules, network-based anomaly detection, and endpoint monitoring are valuable — but they don't catch the failure modes specific to AI systems: prompt injection, model drift, behavioral manipulation, or jailbreak attempts at scale.

Effective AI monitoring covers three layers: model behavior, application integration, and infrastructure.

What to monitor

Model layer

- Significant shifts in output patterns, confidence scores, or response quality
- Unexpected output types or formats not consistent with the model's intended function
- Prompt injection indicators in user-supplied inputs (especially for LLM-powered applications)
- Attempts to extract system prompts or model configuration

Application integration layer

- Unusual data access patterns by AI-integrated services
- High-volume API calls that deviate from baseline usage
- AI outputs being passed to sensitive sinks (database writes, external API calls, command execution)
- Failure rates and error patterns that might indicate adversarial probing

Infrastructure layer

- Unauthorized access to model artifacts or training data storage
- Changes to model configuration, system prompts, or inference parameters
- Unexpected egress to external AI APIs not in your approved inventory

Writing AI policy that teams will actually follow

Most AI security policies fail for the same reason most security policies fail: they're written for compliance, not for usability. A policy that developers can't quickly apply to their decisions will be ignored, not because developers don't care about security, but because ambiguous guidance in a fast-moving environment defaults to inaction or workarounds.

Core components of an AI use policy

A functional AI use policy doesn't have to be long. It needs to be clear, specific, and actionable. At minimum, cover:

- Approved AI tools and models: maintain a list of pre-approved AI tools that teams can adopt without additional review. Make it easy to find and easy to update.
- Review and approval process: define the process for requesting approval for AI tools not on the approved list. Keep it lightweight for low-risk cases; reserve deeper review for Tier 2 and Tier 3 assets.
- Data handling rules: specify which data categories can and cannot be processed by AI systems, and distinguish between internal AI (your own infrastructure) and external AI (third-party APIs and hosted models).
- AI-generated code requirements: clarify that AI-generated code is subject to the same security review requirements as human-written code — including mandatory scanning for vulnerabilities and secrets.
- Disclosure obligations: require that AI integrations be declared in architecture reviews and included in application threat models.
- Acceptable use: be explicit about prohibited uses — training models on regulated customer data without approval, using AI to circumvent access controls, deploying AI agents with permissions beyond what's documented, etc.

Governance structure

Policy without ownership doesn't get enforced. Assign clear accountability:

- AI Security Owner: responsible for maintaining the approved AI inventory, reviewing new AI deployments, and escalating high-risk cases
- Development Team Responsibilities: declaring AI tool use, following data handling rules, submitting AI-generated code for security review
- Procurement and Legal: reviewing vendor contracts for data protection terms; flagging AI capabilities in new SaaS agreements
- Executive Visibility: for Tier 3 AI deployments, ensure leadership is informed and has signed off on risk acceptance

Enforcement through tooling

The most durable security policies are enforced by tooling, not by memory. Where possible, build policy requirements into your existing workflows:

- SAST and SCA scanning in CI/CD that applies to AI-generated code
- Network controls that block egress to unapproved AI endpoints
- SBOM generation for AI components in your build pipeline
- Secret scanning that covers API keys for AI services
- IAM policies that restrict AI service accounts to documented minimum permissions

Four stages. One clear roadmap.

The AI Security Maturity Model is a practical, standards-aligned framework for assessing your AI security posture and building a defensible roadmap. It maps to NIST AI RMF, OWASP AIMA, ISO/IEC 42001, and the EU AI Act — so your maturity progress translates directly to compliance progress.

Most organizations today sit at Stage 1 or 2. That's not a failure — it reflects how fast AI adoption has outpaced governance. The model gives you a clear picture of where you are and what the next step looks like.

1
Emerging

Ad Hoc / Awareness

No formal AI governance or inventory. AI use is experimental, unmanaged, and unmonitored.

2
Developing

Defined / Reactive

Partial AI inventory, initial policies, reactive control mechanisms in place.

3
Controlling

Managed / Proactive

Comprehensive AI governance, formalized processes, continuous risk management.

4
Leading

Optimized / Adaptive

AI assurance embedded across SDLC. Continuous testing and compliance demonstrable.

Stage 1: Emerging
AD HOC / AWARENESS

This is where most teams are today. AI is being used, but governance hasn't caught up. The risk isn't that you're using AI — it's that you don't have full visibility into what's running, who owns it, or what it's connected to.

What this looks like:

- No formal AI inventory — teams don't know what AI is running
- AI use is experimental; shadow AI proliferates without security review
- No governance policies or defined ownership for AI security
- Basic risk awareness only — no formal assessment conducted
- Prompt injection and data leakage risks are unaddressed

Key first steps:

- Create an AI Bill of Materials (AI-BOM) — know what you have
- Assign accountability: who owns AI security?
- Begin basic risk assessments for each AI system in use

Stage 2: Developing
DEFINED / REACTIVE

Policies exist, but enforcement is inconsistent. You've made progress on visibility, but gaps remain — especially around shadow AI, access controls, and integration into the development pipeline.

What this looks like:

- Partial AI inventory exists, but gaps remain (shadow AI persists)
- AI use and security policy drafted, but not consistently enforced
- Human review of AI outputs in some workflows, not all
- Foundational model scanning and dependency checks in place
- Manual monitoring — no systematic model validation yet

Key priorities:

- Standardize AI guardrails: system prompt hardening, input/output validation
- Enforce access controls and encryption across AI model assets
- Integrate AI risk checks into SDLC and CI/CD pipelines

Stage 3: Controlling
MANAGED / PROACTIVE

Formal controls are in place and partially automated. You have a real governance program — but you're not yet doing continuous adversarial testing or quantified risk scoring. The gap between Controlling and Leading is largely about depth and automation.

What this looks like:

- Comprehensive AI governance program with documented processes
- Continuous AI risk detection: prompt injection, data leakage monitoring
- System prompt detection and hardening in place
- Policy enforcement for AI model usage across teams
- Some risk testing, but limited adversarial depth

Key priorities:

- Add AIWE (AI Weakness Enumeration) scoring for quantified, comparable risk metrics
- Move from periodic to continuous adversarial testing
- Establish AI runtime monitoring to detect behavioral drift and hallucination

Stage 4: Leading
OPTIMIZED / ADAPTIVE

AI assurance is embedded across the full SDLC and governance structure. You can answer board and regulator questions with documented evidence — and security is enabling AI adoption velocity, not blocking it.

What this looks like:

- AI assurance embedded across the full SDLC and governance lifecycle
- Automated, continuous red teaming and remediation for AI agents
- Dynamic AI risk scoring (AIWE) with adaptive feedback loops
- AI model explainability, bias assessments, and ethical oversight in place
- Compliance demonstrable — audit-ready documentation and reports

What Leading organizations can do:

- Answer board and regulator questions with documented evidence
- Justify AI security investment with measurable risk reduction metrics
- Accelerate AI adoption — security enables velocity, not blocks it

From emerging to leading: your roadmap

Each stage transition maps to specific NIST AI RMF, OWASP AIMA, ISO 42001, and EU AI Act requirements. Use this table to prioritize action, justify investment, and communicate progress to leadership.

Transition	Priority	Key Actions	Business Outcome
Emerging → Developing	Visibility first	Deploy AI-BOM, assign ownership, run initial threat model	Know what AI you have and who is responsible
Developing→ Controlling	Automate guardrails	System prompt hardening, CI/CD AI checks, policy enforcement	Consistent protection without slowing development
Controlling → Leading	Continuous validation	Automated red teaming, AIWE scoring, runtime monitoring	Prove safety continuously — ready for audits

TAKE THE SELF-ASSESSMENT

Get your personalized AI maturity score in under 5 minutes. Answer 25 questions across 4 control domains, receive your maturity stage score, and get a gap analysis mapped to NIST, OWASP, ISO, and EU AI Act. Visit **mend.io/ai-security-survey**

Aligning governance to frameworks and emerging regulations

The AI regulatory landscape is evolving rapidly. This appendix summarizes the major frameworks and regulations relevant to AI security governance as of early 2026, and maps how the pillars in this guide relate to each.

NIST AI risk management framework (AI RMF)

The NIST AI RMF organizes AI risk management into four functions: Govern, Map, Measure, and Manage. The governance activities in this guide map primarily to the Govern and Map functions — establishing accountability, conducting risk classification, and maintaining inventory. Organizations using the AI RMF can use this guide's maturity model as a practical implementation layer.

EU AI Act

The EU AI Act introduces risk-based obligations tiered by AI use case — with the highest obligations for high-risk AI applications (hiring, credit scoring, biometric identification, etc.). The risk classification framework in Section 2 of this guide is compatible with the Act's risk tier structure. Organizations with EU exposure should ensure their highest-risk AI deployments have conformity assessments and human oversight mechanisms as required by the Act.

ISO/IEC 42001

ISO/IEC 42001 is the international standard for AI management systems, providing a governance framework analogous to ISO 27001 for information security. Organizations pursuing 42001 certification will find that the inventory, risk classification, policy, and monitoring elements of this guide align with the standard's requirements.

Executive order on safe, secure, and trustworthy AI (US)

The US Executive Order on AI (October 2023) established reporting requirements for large AI models and directed agencies to develop sector-specific guidance. For private organizations, the most relevant elements are the emphasis on AI safety testing, transparency about AI capabilities in government-adjacent contexts, and supply chain security — all of which are addressed in this guide.

Mend.io

Mend.io is a leading application security solution that helps organizations fix less and reduce risk faster. Built for both AI-driven and modern development workflows, Mend.io gives teams visibility into all code – human-written, AI-generated, open source, third-party and container components – and helps them prioritize and remediate the risks that matter most.