AI SECURITY ENGINEERING HANDBOOK · 2026

Chapter 01 · AI System Inventory

Standalone study module for LMS delivery and required reading.

FORMAT	USE	SCOPE	AUDIENCE
Standalone PDF	Study module	Single chapter	Learners

CHAPTER 01

AI System Inventory

HANDBOOK STUDY COMPANION: STUDY FRAME

Use this chapter to build vocabulary, judgment, and role-readiness. Pair it with the Field Guide when you need applied actions, checklists, and control execution.

STUDY FOCUS

STUDY FOCUS	WHY IT MATTERS
How to define AI systems, enumerate model and provider dependencies, assign ownership, tier risk, and keep inventory current.	Every control, review, incident response action, and governance claim depends on knowing which AI systems exist and who owns them.

Study Outcomes

- Explain what belongs in an AI system inventory.
- Describe risk tiering criteria for AI-enabled systems.
- Connect inventory records to release gates and evidence.

DOMAIN MAPPING

RELATED AIPSA DOMAINS	APPLIED NEXT STEP	WORKBENCH INSTRUMENTS	RELATED SERVICES
AI Security Foundations	Field Guide foundations	Threat Canvas, Surface Scanner	AI Security Sales Enablement

CERTIFICATION AND ASSESSMENT BOUNDARY

This chapter supports training, diagnostic preparation, scorecards, interviews, and role-readiness evaluation. It does not guarantee credential outcomes.

Every AI security decision — threat model, vendor review, release gate, incident scope — depends on knowing what systems exist and who owns them. Organizations that cannot enumerate their AI deployments make security decisions against a fiction. The threat model targets a system that has since changed. The vendor review covers the provider the team knows about, not the one added last Tuesday. Incident scope gets set by assumption because the retrieval index wasn't in anyone's inventory. Inventory is not a compliance artifact. It is the operational prerequisite for every other AI security control.

Inventory is not a compliance artifact. It is the operational prerequisite for every other AI security control.

HANDBOOK

CORE CONCEPTS

AI SYSTEM ENUMERATION

An AI system is any product feature, internal tool, research deployment, API integration, or vendor capability that uses a model to generate, classify, retrieve, decide, or act. Each system should have a distinct inventory record that includes: system name, owner, business purpose, user population, deployment environment, model provider, model name and version, retrieval index if applicable, agent tools if applicable, data categories, risk tier, and current deployment status. One record per distinct AI-enabled feature or system, not one record per product. A product with multiple AI-powered features at different risk tiers needs multiple records.

MODEL AND PROVIDER DEPENDENCY TRACKING

Each record maps which model and which provider the system depends on. Provider dependency matters for vendor risk assessment, incident scoping, and regulatory obligations. Model version matters because provider-side model updates can change system behavior without any application code change. A system using a self-hosted fine-tuned model and a system using a managed API have different supply-chain risk profiles, different vendor review requirements, and different behavioral monitoring needs.

RISK TIERING

Not every AI system requires equal control depth. Risk tiering assigns each system to a tier — high, medium, or low — based on data sensitivity, action authority, user population, regulatory scope, and reversibility of output effects. Tiering determines which release gates apply, which vendor assessment depth is required, which monitoring requirements are mandatory, and what evidence is expected before deployment. Without tiering, every system either drowns in process or slips through lightweight review. The decision is whether this system's failure would cause significant harm to users, data, or the business.

INVENTORY CONNECTED TO DEPLOYMENT WORKFLOW

Inventory is only as current as the process that updates it. The intake workflow should connect to procurement review, security intake, and release gates so that a new AI system cannot reach production without creating an inventory record. Trigger points include: provisioning a new model provider API key, adding an external model API to a product, creating a retrieval index for production use, connecting an agent to new tool integrations, and changing a system's risk tier due to new capabilities. Inventory updated only at launch will be stale within weeks.

SHADOW AI DISCOVERY

Shadow AI is AI capability deployed without going through security intake — browser AI extensions, SaaS vendor AI add-ons, personal API keys used in production pipelines, low-code model integrations, and AI features in tools that were procured for other purposes. Discovery requires cloud billing review for model API traffic, procurement log analysis, engineering self-disclosure programs, and network monitoring for outbound traffic to known model provider

endpoints. Shadow AI discovered during an incident costs far more than shadow AI discovered through a proactive program.

THE PRACTITIONER'S CHALLENGE

The political challenge is that AI deployment is often framed as engineering velocity, and inventory intake is framed as process overhead. Teams that provision model APIs the same way they provision cloud services do not see a meaningful difference between adding a database and adding a language model endpoint. The practitioner must demonstrate why these are different: a language model endpoint creates a data flow to a third party, a potential training relationship with customer data, a new behavioral dependency that can change without a code deploy, and a vendor relationship with its own breach notification obligations.

The structural challenge is ownership ambiguity. Product engineering ships the AI feature. ML platform provides the inference infrastructure. Procurement approves the vendor. Privacy reviews the data processing. Security owns the intake process. GRC owns the governance evidence. Each of these functions may believe someone else is responsible for maintaining the inventory record. The intake workflow must explicitly assign inventory ownership to a named team with a named trigger, or the record will be incomplete within a release cycle.

The technical challenge is velocity and opacity. AI systems change faster than traditional IT assets. A model provider can update their hosted model without a new API version. An agent can gain new tools without a new deployment. A retrieval index can ingest new document categories without a schema change. The inventory program must define what constitutes a meaningful change that triggers an inventory update, not just what constitutes a new system.

HOW TO APPROACH IT

- Start by enumerating what already exists. Run a discovery sprint before building intake processes. Pull cloud billing records for model provider API calls. Search engineering chat for API key sharing or model provider mentions. Survey product teams about AI-powered features currently running. Review the vendor list for AI and ML services. The first inventory may be incomplete — that is expected. The goal is to establish a baseline that is accurate enough to scope an incident.

- Define a structured record format and require it for every system. A minimal record contains: system name, owner email, business purpose, user-facing or internal classification, deployment environment, model provider name, model name and version, data categories processed, risk tier, retrieval index existence, agent tool list if applicable, and evidence links. Make the record template available in the tools teams already use for project tracking. A record in the tool teams use is more likely to be maintained than a record in a separate system no one checks.

- Build intake as a gate, not a form. The intake workflow should fire when a new model API key is provisioned, a new AI vendor is added to the approved list, a new retrieval index is built for production, or an agent is connected to new external tool integrations. Intake approval should be a prerequisite for production deployment, not an after-the-fact review. Connect the intake completion status to the release gate so that a system with incomplete intake cannot pass the release checklist.

- Apply risk tiering immediately. Assign each system a tier — high, medium, or low — based on data sensitivity, action authority, and user population. High-tier systems require depth: full threat model, vendor security assessment, eval evidence before every model version change, and telemetry review. Medium-tier systems require standard review and annual re-assessment. Low-tier systems require basic intake and change notification. Document the tiering criteria so teams can self-assess new systems before security review.

- Build shadow AI discovery as a continuous program, not a one-time audit. Quarterly reviews of cloud billing and procurement for new model API traffic, engineering-facing self-disclosure with low friction and no penalty, and network monitoring for outbound traffic to known model provider endpoints are the minimum components. The goal is completeness, not punishment. A team that discloses an unregistered AI tool and goes through intake promptly is a success story. A team that hides a tool because intake feels punitive is the failure mode.

OUTPUTS AND DELIVERABLES

- The foundational artifacts are the **AI system inventory template**, **intake workflow specification**, and **risk tiering rubric**. The inventory template defines the required fields for a complete record, the optional fields for high-tier systems, and the evidence links section that connects the record to downstream control artifacts. The intake workflow specification names the trigger events, required approvals, and release gate connection. The tiering rubric defines high, medium, and low criteria with decision-useful examples from the organization's actual AI footprint.

- The operational artifacts are the **intake request process**, **discovery sprint playbook**, and **shadow AI disclosure path**. The intake request process gives engineering teams a clear sequence: submit the intake record, receive a risk tier determination, complete required controls for that tier, and receive production approval. The discovery sprint playbook defines the quarterly shadow AI review: what sources are checked, who runs it, how findings are triaged, and how new systems enter intake rather than getting reported as violations. The disclosure path gives teams a low-friction way to bring unregistered tools into the program.

- The governance artifacts are the **inventory reporting dashboard**, **stale record review schedule**, and **AI asset register integration with vendor management**. The reporting dashboard shows inventory coverage, tiering distribution, systems with missing evidence, and systems pending intake approval. The review schedule defines when each record must be re-verified — high-tier systems quarterly, medium-tier annually, with automatic triggers on model version changes. The vendor management integration ensures that every AI vendor in inventory is also reflected in the vendor risk program.

COMMON FAILURE MODES

- **One-Time Inventory:** The organization runs a discovery sprint, produces a snapshot inventory, and never updates it. Within two release cycles, the inventory is materially incomplete. Prevent this by connecting inventory updates to the deployment workflow rather than treating inventory as an annual compliance task.
- **Product-Level Granularity:** The team registers products rather than features, resulting in one inventory entry for a product with three AI-powered features, two model providers, an embedded retrieval index, and an agent with four tools. The inventory appears complete while the actual security surface area is invisible. Require feature-level records for any product with multiple distinct AI capabilities.
- **No Shadow AI Program:** The intake process handles new systems but has no mechanism to discover what bypassed intake. Each quarter, the shadow AI footprint grows. Prevent this by treating discovery as a continuous program with defined cadence, not a one-time exercise.
- **Inventory Without Evidence Links:** The records exist but do not link to the security artifacts — threat models, eval results, vendor assessments, telemetry dashboards — that prove controls operate. The inventory becomes a registry of systems rather than a governance artifact. Require evidence links as part of record completion for high-tier and medium-tier systems.

IMPLEMENTATION CHECKLIST

- [ ] Define the inventory record template with required fields for each risk tier.
- [ ] Build intake as a deployment gate that fires on defined trigger events.
- [ ] Complete a discovery sprint to establish a baseline inventory before improving the intake process.
- [ ] Define risk tiering criteria and assign a tier to every existing system.
- [ ] Create a shadow AI disclosure path with no penalty for teams that self-report.
- [ ] Connect inventory records to vendor management for all external model providers.
- [ ] Define a review schedule for stale records with automatic triggers on model version changes.
- [ ] Integrate inventory reporting into security governance reviews.

RELATED READING

- Handbook chapters: Chapter 14 (Governance Evidence and Customer Trust) for connecting inventory to control evidence; Chapter 8 (Model and Provider Risk) for vendor dependency records; Chapter 9 (AI Supply Chain) for model artifact registry connection.
- Field Guide: AI Security Foundations for inventory checks, trust mapping, owner records, and evidence requests.

AI System Inventory AISECURITY.LLC