AI PRODUCT SECURITY IN THE AGE OF MYTHOS · 2026

Chapter 06 · Inventory Is the First Control

Standalone reading module for LMS delivery and required reading.

FORMAT	USE	SCOPE	AUDIENCE
Standalone PDF	Required reading	Single chapter	Learners

CHAPTER 06

Inventory Is the First Control

82:1

MACHINE IDENTITIES

CyberArk reported 82 machine identities for every human identity, with 42% of machine identities holding privileged or sensitive access.

CYBERARK, 2025

You cannot secure an AI product whose authority graph you cannot draw.

Inventory in AI product security is not clerical work. It is the first control because AI systems connect models, prompts, data, tools, identities, secrets, agents, logs, and human approval paths. A team that cannot name those connections cannot reason about blast radius, trust boundaries, or release gates.

The Authority Behind the Interface

The dangerous system rarely introduces itself as dangerous.

It may arrive as a support assistant that summarizes customer tickets and drafts replies. The value proposition is simple: speed up response time, improve consistency, reduce manual work. At intake review, the product looks like a chat surface with a narrow purpose.

The authority is usually hidden behind integration.

The support assistant reads Zendesk to fetch recent tickets. It also reads Slack to fetch internal escalation threads. It indexes the Drive folder containing product FAQs, but that folder also contains archived incident reports and confidential customer communications. The retrieval system does not distinguish between public help content and internal notes. It ranks by similarity, not by authorization. The service token driving the retrieval has permissions to read the entire Drive, not just the FAQ folder. The system calls a workflow automation tool that can send emails to customers. The tool has permissions to update CRM records. The assistant stores conversation history in a memory vector store, indexing by customer ID. It retrieves prior interactions across all sessions. There is no tested kill switch. Disabling the chat UI does not stop background indexing.

The support team works with the system for a month. It becomes part of their workflow. Someone asks: can it handle escalations? A small workflow rule is added: if the assistant flags high-priority issues, send an email to the team. Another person asks: can it track customer sentiment? A small feature is added to update a CRM field with urgency signals. A third person suggests: can it offer refund recommendations? A threshold-based approval is wired in.

Six months later, a hostile customer notices they can craft a ticket that causes the system to recommend inappropriate refunds. An internal user notices they can see data from other tenants in the assistant's reasoning. An attacker notices the Slack channel contains API keys in "confidential incidents" that the system indexed.

The inventory failure is not that the assistant was missing from a spreadsheet. The failure is that nobody could draw the authority graph—all the connections between data, tools, tokens, and decisions—fast enough to know what to fix first.

This is not unique to AI. Major security breaches reveal the same pattern: organizations had some visibility into applications but were blind to delegated authority. Okta's support-system compromise showed how support systems become privileged identity infrastructure. MOVEit's exploitation chain was made worse by incomplete exposure mapping. Colonial Pipeline's operational disruption came from a single exposed credential. SolarWinds revealed that organizations often inventory software but miss the trust-path inventory—what each component can do to what other component, and under which identity.

Modern inventory fails because modern authority is no longer concentrated in applications. It is distributed across service accounts, OAuth apps, API keys, SaaS connectors, CI/CD tokens, workflow automations, browser sessions, memory stores, and machine identities. CyberArk's 2025 research reports that machine identities now outnumber human identities 82:1, and expects AI to become the largest creator of new privileged or sensitive identities in 2025. That changes what inventory means. A product-security inventory that names only applications is blind to the non-human authority actually moving through the environment.

A second failure is fragmentation. CyberArk reports that 70% of organizations identify identity silos as a root cause of cybersecurity risk. That maps directly to AI product security. A support assistant may appear once in a product catalog, but its authority may be split across Zendesk, Drive, Slack, CRM, a vector index, a service token, and a workflow automation. The system can be named and still be unknown.

Most organizations inventory applications. Far fewer inventory delegated authority.

AI product inventory exists to make hidden authority visible before it becomes incident scope.

Catalog Versus Control

An AI system can be accurately named and still be dangerously unknown.

"Support Assistant v1.2" may be present in the inventory. The inventory is useless if it does not answer:

- Which data sources does the retrieval system index? (Zendesk, Slack, Drive, CRM?)
- Which permissions are required to retrieve that data? (User scoped? Tenant scoped? Full read?)
- Does the retrieval system respect source-level ACLs? (Does it check permissions before including a chunk in context?)
- What is the version of the retrieval index? (Is it up-to-date with current permissions?)
- Which tools can the system invoke? (Send email? Update CRM? Open support cases?)
- What credentials does each tool use? (Service token? User token? OAuth?)
- Which tools require human approval? (Which actions pause and ask before executing?)
- How is approval evidence logged? (Can an incident responder reconstruct why a decision was made?)
- Which outputs reach customers? (Can hostile input influence external-facing decisions?)
- How is conversation history stored? (Is it encrypted? Tenant-scoped? Can the team delete it?)
- How is the system disabled? (Does disabling the chat UI stop all background processes?)
- Who owns this system? (Can they force a shutdown?)

A catalog can tell you the assistant exists. A control-grade inventory can tell you whether the assistant can read customer escalations, whether those escalations are tenant-scoped, whether the tool token can write to CRM, whether a human approves outbound messages, whether the logs show retrieved chunks, and who can shut the system down.

> ***The difference between catalog and control:*** *A catalog tells you the system exists. A control-grade inventory tells you whether the organization can respond to an incident or just document it.*

The Seven-Question Authority Graph

If the team cannot draw the authority graph, it cannot responsibly approve launch.

A useful inventory record should let a reviewer answer seven questions clearly:

1. What data can enter context? Which data sources does the system retrieve from? Databases, file systems, external APIs, other models' outputs, user input, logs? Is all data equal or is some data sensitive? Can the system distinguish?

2. Which model or provider receives it? Which AI model runs the inference? Whose model is it? Which version? Are weights frozen or fine-tuned? Who controls the prompt? Can users modify the system message?

3. Which tools can the system call? What is the API surface? Can the system send email? Update records? Create cloud resources? Query databases? Approve transactions? Each tool capability expands the blast radius.

4. Which identity does each tool use? Does the tool use a service token, user token, or OAuth? How broad are the permissions? Can the tool write to only the data it should, or does it have write access to more?

5. Which actions require approval? Which tool calls are automatic, and which require human review? Does "approval" mean a button click with no context, or a detailed review? Can the approver see what data was retrieved and why?

6. Which logs can reconstruct the decision? After an incident, can the team see: what data was retrieved, from which source, which version, which model, which prompt, what output was generated, which tool was called, and what actually changed in the system? Or are parts of the flow invisible?

7. Who can disable the system? What is the kill-switch procedure? Is it tested? Does disabling the UI stop all backend processes, or does indexing/batch processing continue? Can the owner force a shutdown without waiting for the next release cycle?

If the team cannot answer all seven clearly, the system is incompletely inventoried.

An inventory also needs to know which external claims govern the system: customer contracts, trust-center statements, privacy commitments, and AI disclosures.

In the support-assistant example from the preface, the authority graph is the difference between "chatbot for tickets" and an inspectable system: ticket data, knowledge-base retrieval, CRM writes, billing-credit requests, approval gates, service identity, logs, and kill switch all appear in one reviewable record.

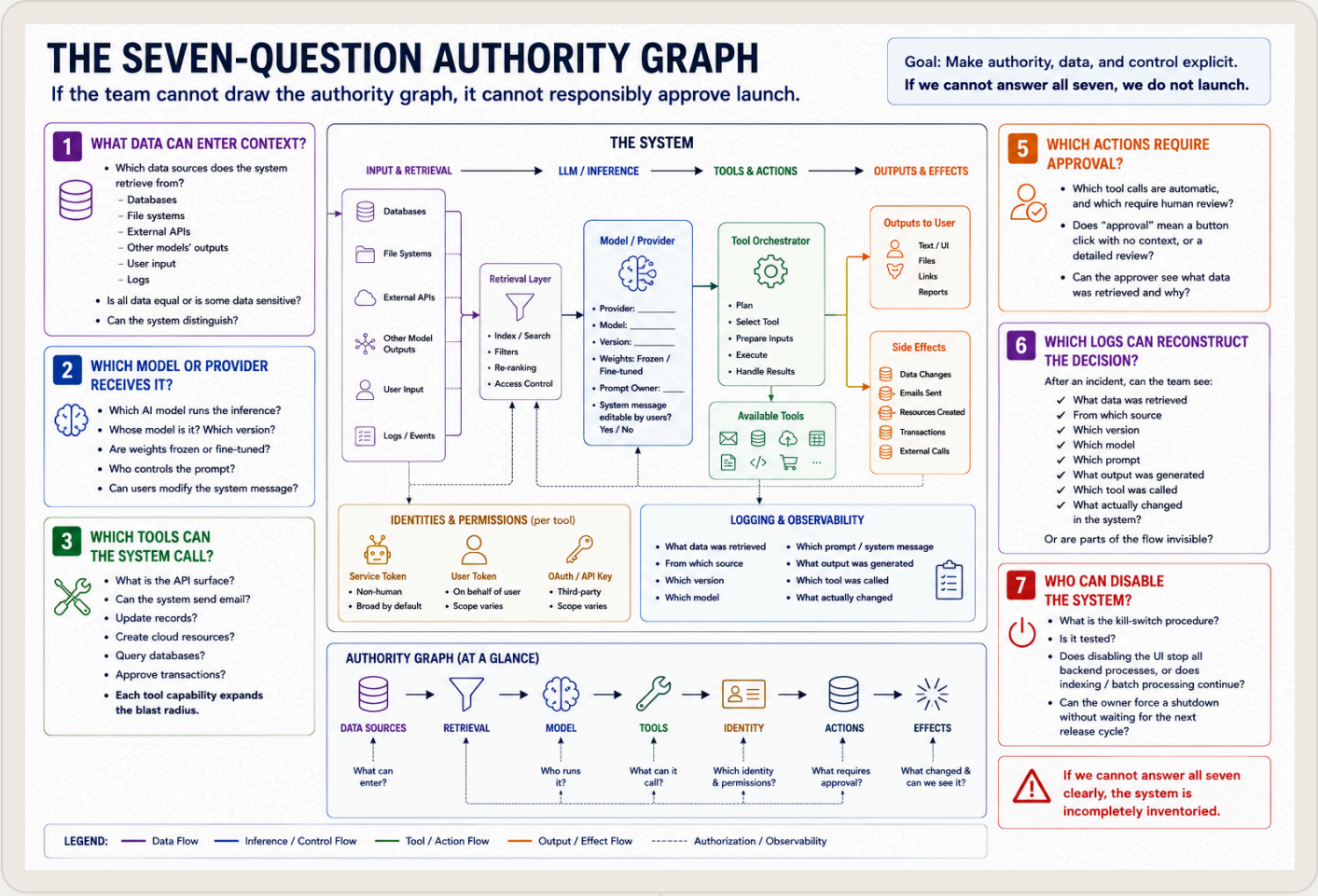

FIGURE 1: IF THE TEAM CANNOT DRAW THE AUTHORITY GRAPH, IT CANNOT RESPONSIBLY APPROVE LAUNCH. THE SEVEN-QUESTION FRAMEWORK MAPS DATA SOURCES THROUGH RETRIEVAL AND MODEL INFERENCE INTO TOOLS, IDENTITIES, APPROVALS, LOGS, SIDE EFFECTS, AND SHUTDOWN CONTROL.

The Blind Spots Reality

Inventory failures cluster around three shadows:

Shadow systems – The agent that runs nightly in a cron job no one remembers creating. The "experimental" chatbot a team deployed to Slack six months ago. The internal dev tool with a broad API token that got copied three times. Organizations rarely see these until incident response has to trace backwards from the damage.

Authority creep – The assistant launched read-only. Then it needed to update a status field. Then it needed to send notifications. Then it needed to call a refund workflow. Each addition made sense at the time. Together they created a system whose actual authority is invisible to the team running it. The owner knows what the system was supposed to do. They often do not know what it can actually do.

Fragmented identity – The system is "named" but its authority is scattered. Service token in Vault, OAuth app in GitHub, read permissions in Sheets, write permissions in a SaaS connector. One service account was copied from a staging template and never audited. A second token is shared between three different tools. The retrieval index is version 17 but the permission list is from version 14. The organization has an inventory row for "Customer Support Assistant v1.2" but nobody can draw a line from that name to all the credentials and permissions that actually execute under that name.

> ***The real inventory failure:*** *It is not having no catalog. It is having a catalog but it is completely disconnected from the actual authority moving through the system.*

Ownership and Living Inventory

Naming an owner turns inventory from artifact into control. The owner must know the seven questions—before launch and as the system changes. They own the risk. They own the response.

But ownership only matters if inventory stays current. A new tool, a new data source, a permission change, a token rotation—all require the inventory to move. Inventory that stops updating is obsolete.

The reality: ownership often assumes a stable system. AI systems do not stabilize. Prompts change. Models update. Tools accumulate. Context windows shift. Teams iterate without realizing they are shifting the authority surface. The owner's job is to keep the seven questions answerable even as the answers change.

The next chapter will show how that visibility changes when the system's authority becomes a moving target. Today's inventory is tomorrow's constraint—one that will change as the model updates, tools are added, and data sources evolve.

Detailed inventory templates, ledger schemas, common blind spots, and authority graph examples – in Appendix B.

Sources

- CyberArk 2025 Identity Security Landscape: https://www.cyberark.com/press/machine-identities-outnumber-humans-by-more-than-80-to-1-new-report-exposes-the-exponential-threats-of-fragmented-identity-security/
- NIST AI RMF: https://www.nist.gov/itl/ai-risk-management-framework
- NIST SSDF SP 800-218: https://csrc.nist.gov/pubs/sp/800/218/final